

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh a implementace databázové aplikace pro dodavatelskou firmu
Design and Implementation of Database Application for the Supplier Company

Student: Aleš Miňo

Vedoucí bakalářské práce: Ing. Vítězslav Novák, Ph.D.

Ostrava 2015

Zadání bakalářské práce

Student:

Aleš Miňo

Studijní program:

B6209 Systémové inženýrství a informatika

Studijní obor:

6209R025 Systémové inženýrství a informatika

Téma:

Návrh a implementace databázové aplikace pro dodavatelskou firmu
Design and Implementation of Database Application for the Supplier
Company

Zásady pro vypracování:

1. Úvod
 2. Teoretické pojmy a východiska tvorby databázových aplikací
 3. Analýza současného stavu ve firmě
 4. Návrh a implementace databázové aplikace
 5. Závěr
- Seznam použité literatury
Seznam zkratk
Prohlášení o využití výsledků bakalářské práce
Seznam příloh
Přílohy

Seznam doporučené odborné literatury:

CONOLLY, T., R. HOLOWCZAK a C. BEGG. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.
KRUCZEK, Aleš. *Microsoft Access 2010: podrobná uživatelská příručka*. Brno: Computer Press, 2010. 392 s. ISBN 978-80-251-3289-0.
SHEPHERD, Richard. *Access VBA: výukový průvodce*. Brno: Computer Press, 2012. 397 s. ISBN 978-80-251-3686-7.

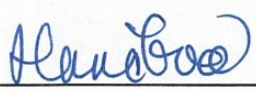
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

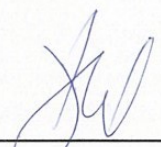
Vedoucí bakalářské práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: 21.11.2014

Datum odevzdání: 07.05.2015



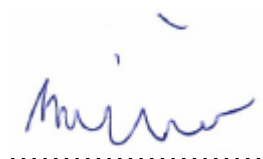

doc. Ing. Jana Hančlová, CSc.
vedoucí katedry


prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Místopřísežné prohlášení o samostatném vypracování bakalářské práce

„ Prohlašuji, že jsem celou práci, včetně všech příloh vypracoval samostatně“.

Zároveň bych na tomto místě chtěl poděkovat svému vedoucímu bakalářské práce, Ing. Vítězslavu Novákovi, PhD. za vedení bakalářské práce, cenné rady a připomínky. Dále také své rodině, která mi umožnila studovat.



Aleš Miňo

Datum odevzdání bakalářské práce: 7. května 2015

Obsah

1	Úvod	6
2	Teoretické pojmy a východiska tvorby databázových aplikací.....	8
2.1	Vysvětlení základních pojmů	8
2.1.1	Data	8
2.1.2	Informace	8
2.1.3	Databáze.....	8
2.1.4	Databázová aplikace.....	8
2.2	Databázový model.....	9
2.2.1	Hierarchický model	9
2.2.2	Síťový model.....	10
2.2.3	Relační model	11
2.2.4	Objektově orientovaný model.....	12
2.2.5	Objektově relační model	12
2.3	Systém řízení báze dat.....	12
2.4	Datové modelování.....	13
2.4.1	Entitně – Relační modelování	13
2.4.2	Konceptuální model	14
2.4.2.1	Entita	14
2.4.2.2	Atribut	15
2.4.2.3	Vztahy	15
2.4.2.4	Klíče	17
2.4.3	Logický model	19
2.4.3.1	Tabulka.....	19
2.4.3.2	Normalizace	20
2.4.3.3	Omezení.....	21

2.4.4	Fyzický model	22
2.5	SQL	23
2.5.1	Kategorie příkazů	23
2.5.1.1	DDL	23
2.5.1.2	DML	23
2.5.1.3	DCL	24
2.5.1.4	TCC	24
2.6	Microsoft Access	24
2.6.1	Tabulky	25
2.6.1.1	Vytvoření tabulky	25
2.6.1.2	Přehled datových typů	25
2.6.2	Dotazy	26
2.6.3	Formuláře	27
2.6.3.1	Visual Basic For Application (VBA)	27
2.6.4	Sestavy	28
3	Analýza současného stavu ve firmě	29
3.1	O společnosti	29
3.2	Současná situace	29
3.3	Požadavky	29
4	Návrh a implementace databázové aplikace	31
4.1	Konceptuální návrh databáze	32
4.2	Logický návrh databáze	33
4.3	Fyzický návrh databáze	34
4.4	Návrh tabulek	34
4.4.1	Tabulka tblOdberatel	35
4.4.2	Tabulka tblDodavatel	36

4.4.3	Tabulka tblProdukt	37
4.4.4	Tabulka tblObjednavky	37
4.4.5	Tabulka tblDodavky	38
4.4.6	Tabulka tblPolozkyObjednavky	39
4.4.7	Tabulka tblPolozkyDodavky	40
4.5	Návrh formulářů a sestav	40
4.5.1	Navigační formulář frmUvod	41
4.5.2	Formulář frmProdukt	41
4.5.3	Formulář frmDodavatel	43
4.5.4	Formulář frmOdberatel	44
4.5.5	Formulář frmObjednavka	45
4.5.6	Formulář frmDodavky	48
4.5.7	Sestavy	49
5	Závěr	51
	Seznam použité literatury	51
	Seznam zkratk	53
	Prohlášení o využití výsledků bakalářské práce	
	Seznam příloh	
	Přílohy	

1 Úvod

Nezbytnou podmínkou úspěšné podnikatelské činnosti je umět pracovat s množstvím dat, které je objemnější než kdy předtím. Schopnost využít tyto data ve svůj prospěch nám zajišťují moderní informační technologie, ke kterým začíná inklinovat většina podnikatelů.

Tématem, kterým se tato bakalářská práce zabývá, je vytvoření databázové aplikace pro dodavatelskou firmu, která doposud žádným databázovým systémem nedisponuje, proto je důležité klást důraz na jednoduchost a přehlednost. Firma dováží zboží ze zahraničí a následně ho skladuje v České republice, kde ho distribuuje svým odběratelům. S narůstajícím počtem zboží přibývá stále více problémů spojených s řízením prodeje. Proto je nutné, aby databázová aplikace sloužila především k zjednodušení a zrychlení jejich podnikatelské činnosti, z čehož vyplývá potenciální zvýšení hospodářského výsledku.

Cílem bakalářské práce je vytvořit databázovou aplikaci v programu Microsoft Access, která by měla splňovat následující funkce:

- zaznamenávání veškerých údajů o zboží,
- správa dodacího listu a objednávkového formuláře,
- řízení, správa a evidence jednotlivých objednávek a dodávek,
- zaznamenání veškerých údajů o odběratelích a dodavatelích,
- jednoduchý grafický design s intuitivním ovládáním.

Databázová aplikace je určena primárně pro obchodníky, kteří zajišťují celý životní cyklus prodeje, od dovezení zboží, jeho skladování až po jeho distribuci.

Koncept této práce je členěn do pěti základních částí, z nichž první je samotný „Úvod“, kde je stručně popsána problematika, a jsou definovány základní cíle bakalářské práce.

V další části „*Teoretické pojmy a východiska*“, jsou popsány základní pojmy problematiky databázových aplikací, a seznámení s programem Microsoft Access a Visual Basic, kde bude aplikace vytvořena.

Následující dvě části „*Analýza současného stavu ve firmě*“ a „*Návrh a implementace databázové aplikace*“ se budou zabývat praktickou částí bakalářské

práce, a to zjištěním aktuální situace a samotným návrhem aplikace pomocí tabulek, jejich propojením pomocí vztahů, a vytvořením formulářů a sestav nutných pro správný chod aplikace.

Poslední částí je „závěr“, který se zabývá celkovým hodnocením práce, dosažení stanovených cílů či naopak.

2 Teoretické pojmy a východiska tvorby databázových aplikací

Tato kapitola se zabývá teoretickými pojmy, které jsou východisky pro aplikační část této práce. Bude obsahovat základní informace pro tvorbu relačních databází, seznámení s programy Microsoft Access a Visual Basic for Application.

2.1 Vysvětlení základních pojmů

V této podkapitole budou vysvětleny základní pojmy, kterými jsou Informace, Data a Databáze, které by měl v dnešní době znát a dokázat vysvětlit naprosto každý a jsou klíčem k pochopení dané problematiky.

2.1.1 Data

Data jsou fakta, která mají například pro určitou firmu význam, ale doposud neprošla žádným zpracováním, takže je nedokážeme správně interpretovat. (Connolly, Begg a Holowczak, 2009)

2.1.2 Informace

Informace jsou data, které již prošla zpracováním a získala svou strukturu, takže je dokážeme správně interpretovat a tím získávají pro konkrétní firmu přidanou hodnotu. (Connolly, Begg a Holowczak, 2009)

2.1.3 Databáze

Databáze je množina vzájemně uspořádaných a souvisejících dat, která se jeví jako ucelená jednotka, a je uložena na určitém paměťovém médiu. Tato definice je velice široká, z toho důvodu, že mezi jednotlivými produkty a výrobci databázových systémů existuje vysoká variabilita. Například v Microsoft Access se celá databáze ukládá do jediného souboru, takže bychom ji mohli definovat jako určitý soubor, který obsahuje datové položky. (Oppel, 2006)

2.1.4 Databázová aplikace

Databázová aplikace je program, který funguje jako rozhraní a integruje databázi s SQL příkazy, které uživatel pomocí rozhraní zadá. Je naprogramovaná v některém z vyšších programovacích jazyků (Java, C#, VB). V dnešní době se používají zejména online aplikace. (Connolly, Begg a Holowczak, 2009)

2.2 Databázový model

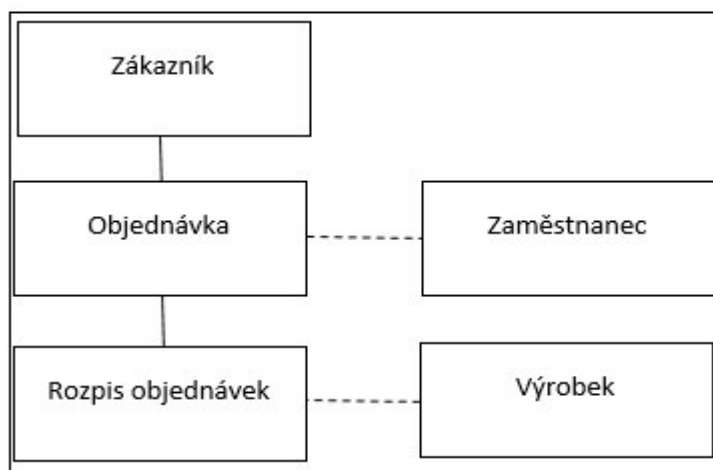
Databázový model je architektura, podle které databázový systém ukládá a provazuje jednotlivé databázové objekty a tím tak vyjadřuje podobu reálného světa. (Oppel, 2006)

2.2.1 Hierarchický model

První databáze byly postaveny na hierarchickém modelu. Jednotlivé záznamy jsou zde propojeny pomocí ukazatelů, které ukazují na příslušné s nimi svázané záznamy. Podle těchto ukazatelů se dá jednoduše dohledat, kde každý záznam fyzicky leží. Každá vazba mezi záznamy funguje na principu rodiče a jeho potomka. Každý rodič může mít libovolný počet potomků, ale potomek má jen jednoho rodiče. Definice jednotlivých záznamů je příliš přísně nastavena, a tak do něj některá data téměř nelze zakomponovat a zachytit situaci reálného světa se stává nemožné. Nejznámější hierarchickou databází býval systém ISM od firmy IBM. (Oppel, 2006)

Na obrázku 2-1 je vidět hierarchický model databáze, který používá záznamy Zákazník, Objednávka, Rozpis, Zaměstnanec a Výrobek. Z obrázku je jasně patrná největší nevýhoda a zároveň největší omezení hierarchického modelu, kvůli kterým se tento model prakticky přestal používat.

Záznamy Zaměstnanec a Výrobek jsou do modelu zapojeny přerušovanou čarou, protože nemůžou být spojeny s danými záznamy pomocí ukazatelů. Žádný záznam nemůže mít více než jednoho rodiče, z čehož vyplývá, že nemůžeme propojit Zaměstnance s Objednávkou, protože záznam objednávka již rodiče má, a to záznam Zákazník. Podobně to funguje u záznamu Výrobek. (Oppel, 2006)



Obrázek 2-1 Ukázka Hierarchického modelu, zdroj: Oppel (2006, s. 25), upraveno

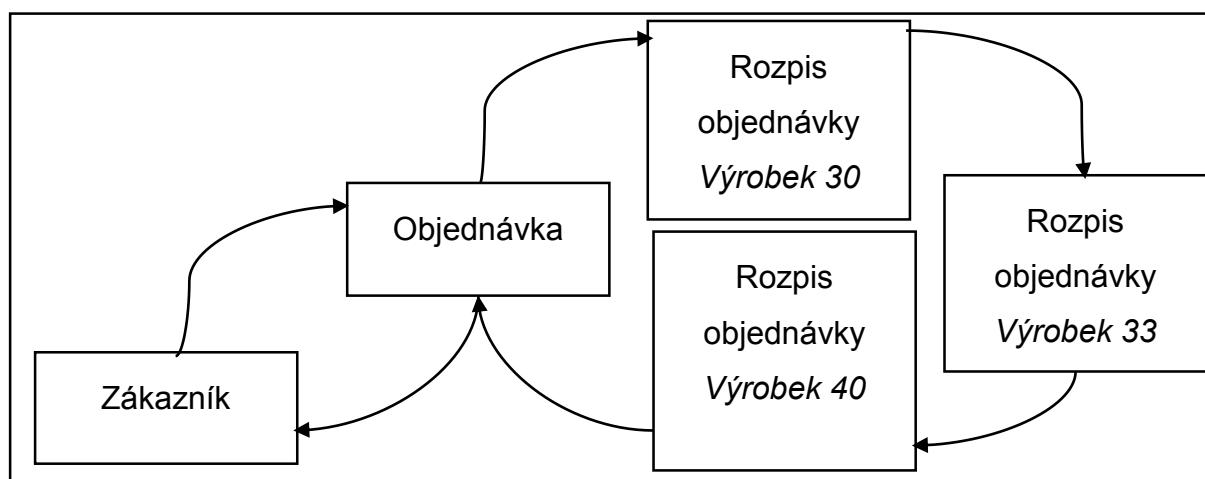
V praxi to znamenalo, že technici museli tyto omezení často obcházet a vytvářet pod každým rodičem jeho kopii, což vedlo k velkému plýtvání kapacity disku, což bylo nepřijatelné. (Oppel, 2006)

2.2.2 Síťový model

Síťový model databází vznikl zhruba ve stejné době jako model hierarchický. Nejpoužívanější databází fungující na tomto modelu byl systém IDMS, od společnosti Cullinane. (Oppel, 2006)

V tomto modelu se příbuzné záznamy propojují také pomocí ukazatelů, a na rozdíl od modelu hierarchického jsou zde jednotlivé vztahy mezi záznamy pojmenovány, a díky tomu se může strana potomků účastnit několika relací, takže programátor může používat přechody z jednoho záznamu na druhý, prostřednictvím konkrétních relací. Síťový model znamenal vyšší použitelnost i flexibilitu pro ukládání záznamů, ale znamenalo to daleko větší složitost. (Oppel, 2006)

Na obrázku 2-2 je vidět základní nevýhoda tohoto modelu a důvod proč se přestal tento model používat. Z obrázku lze vyčíst, že je potřeba získat informace o Výrobku 40, musí se projít přes všechny Výrobky dané objednávkou. Tento obrázek je velice jednoduchý, a už na něm je patrné, že síťový model funguje jen za předpokladu, že všechny ukazatele jsou kruhové nebo cyklické. (Oppel, 2006)



Obrázek 2-2 Ukázka Síťového modelu, zdroj: Oppel (2006, s. 27), upraveno

Proces, který funguje, jako navigace v síťovém modelu je nazýván „procházení množiny“, a jeho úkolem je výběr cest v databázové množině. Při tomto procesu může být vybráno několik cest, nebo se také může dostat do „slepé uličky“, jinými slovy

zacyklit se. Obrovská složitost síťového modelu a jeho náročnost na údržbu byly hlavními důvody jeho zániku. (Oppel, 2006)

2.2.3 Relační model

V historii informačních technologií je považováno vytvoření relačního modelu za jeden ze zlomových momentů, o který se postaral Dr. E. F. Codd svou výzkumnou prací, na jejímž základě se zrodil relační model databází. (Oppel, 2006)

Relační model je postaven takovým způsobem, že dotazy pracují s určitou množinou dat, nikoliv s jednotlivými záznamy jako tomu bylo u modelů síťového a hierarchického. Základní myšlenkou je svázat záznamy podle potřeby, pokud spolu souvisí a na tomto principu vytvářet později dotazy. (Oppel, 2006)

V tomto modelu jsou data zastoupena pomocí dvourozměrných tabulek. Propojování jednotlivých záznamů, které spolu souvisí, se neprovádí pomocí fyzických ukazatelů, ale do jednotlivých tabulek se vloží společný atribut (cizí klíč). (Oppel, 2006)

Na obrázku 2-3 je znázorněn relační model, který obsahuje tabulky Zákazník, Zaměstnanec, Objednávka, Rozpis objednávek a Výrobek. Jednotlivé čáry mezi tabulkami znázorňují typy vztahů. Na tomto obrázku jsou všechny vztahy typu 1:N, to znamená, že například jeden Zaměstnanec se stará o více Objednávek a k jednomu Zákazníkovi může připadat více objednávek. To samé platí například u Rozpisu objednávek, nebo výrobku. (Oppel, 2006)



Obrázek 2-3 Ukázka Relačního modelu, zdroj: Oppel (2006, s. 29), upraveno

Hlavně díky své flexibilitě a schopnosti zachytit reálné situace byl relační model přijat. Je to základ většiny databázových systémů a ještě po mnoho let bude. (Oppel, 2006)

2.2.4 Objektově orientovaný model

Objektově orientovaný model vznikl v 70. letech minulého století, ale používat se začal až o dvacet let později. Hlavním důvodem, proč se začal model používat je fakt, že dřívější databázové modely nebyly schopny pracovat se složitějšími daty, jako jsou obrázky, video nahrávky, atd. (Oppel, 2006)

Objekt znamená logickou skupinu navzájem souvisejících dat, které reprezentují něco z reálného světa, nejčastěji osoby nebo věci. Součástí každého objektu jsou *datové položky*, jinými slovy proměnné jako je například jméno zákazníka nebo rodné číslo. Dalším důležitým pojmem je *metoda*, která reprezentuje určitou část aplikační logiky a pracuje nad vybraným objektem. Hlavní rozdíl mezi objektově orientovaným modelem a ostatními modely je fakt, že k jednotlivým datovým položkám daných objektů lze přistupovat pouze prostřednictvím metod. Tato vlastnost je známá pod názvem *Zapouzdření*. Jednotlivé objekty jsou strukturovány do hierarchií tříd, kde jsou jednotlivé datové položky a metody nadefinovány a jsou dostupné pro všechny třídy v daném modelu. Této vlastnosti se říká *Dědičnost*. (Oppel, 2006)

Objektové programování jako takové původně vypadalo, že nebude mít v budoucnu uplatnění, ale opak se stal pravdou a dostalo se téměř do všech částí moderního informačního světa. (Oppel, 2006)

2.2.5 Objektově relační model

Objektově relační model vznikl na základě faktu, že obyčejný relační model a objektově orientovaný model mají nesporné výhody, které je třeba spojit dohromady. Jednotliví výrobci relačních databází rozšířili svoje relační modely o prvky objektové a pokusili se tak integrovat největší přednosti obou modelů do jednoho celku. Mezi nejznámější objektově relační modely patří Oracle, DB2 a Informix. (Oppel, 2006)

2.3 Systém řízení báze dat

Systém řízení báze dat se označuje anglickou zkratkou DBMS. Je to software, který integruje uživatele s jeho databázovou aplikací a s databází. Hlavní funkcí DBMS

je poskytování řízeného přístupu k databázi. Existuje mnoho výrobců DBMS, mezi nejznámější patří Oracle nebo DB2 od IBM. (Connolly, Begg a Holowczak, 2009)

Systém řízení báze dat (DBMS) poskytuje všechny základní služby, které jsou vyžadovány k údržbě a uspořádání databáze, mezi které patří například:

- Přesun dat z různých datových souborů do jiných datových souborů
- Řízení paralelního přístupu k datům více uživatelů, tak aby nenastala žádná kolize při případných paralelních aktualizacích
- Řízení transakcí, tak aby se provedly všechny změny, které transakce vykonává, v opačném případě pokud nastane chyba a neprovede se žádná změna, tak se data vrátí do původního stavu. Některé relační systémy řízení báze dat transakce nepovolují.
- Podpora dotazovacího jazyka (SQL), který slouží pro práci s daty. SQL je primární dotazovací jazyk, který relační systémy využívají.
- Slouží pro zálohování databáze, a pro obnovení databáze při případném selhání.
- Zajišťuje bezpečnostní mechanismy, které zabraňují neoprávněnému přístupu a změně dat. (Oppel, 2008)

2.4 Datové modelování

Datové modelování je proces, který zobrazuje data na určité úrovni abstrakce s rozdílným pohledem na data v jednotlivých fázích. Od vymezení konkrétních částí reálného světa až po fyzickou implementaci do počítače. Skládá se z konceptuálního, logického a fyzického modelu. (Connolly, Begg a Holowczak, 2009; Riordan, 2000)

2.4.1 Entitně – Relační modelování

Jedním z největších problémů při návrhu databáze je fakt, že databázoví návrháři, programátoři a analytici se dívají na data rozdílným způsobem. Při návrhu databáze je naprosto nutné, aby existoval stejný pohled na data všech zúčastněných, jinak nelze vytvořit databázovou aplikaci, která by pokrývala požadavky reálného světa. ER model není technický a neobsahuje víceznačnosti a proto nám umožňuje komunikaci mezi všemi. (Connolly, Begg a Holowczak, 2009)

ER model návrhu databáze je metoda, která nám umožňuje návrh databáze od shora dolů. ER modelování začíná určením klíčových dat (entity) a vztahy mezi nimi, které je třeba v modelu reprezentovat. Pak se postupuje k dalším podrobnostem, jako jsou atributy entit, které nám říkají, jaké data chceme uchovávat. ER model se při návrhu používá hlavně k vytvoření konceptuálního modelu, a také logického modelu. (Connoly, Begg a Holowczak, 2009)

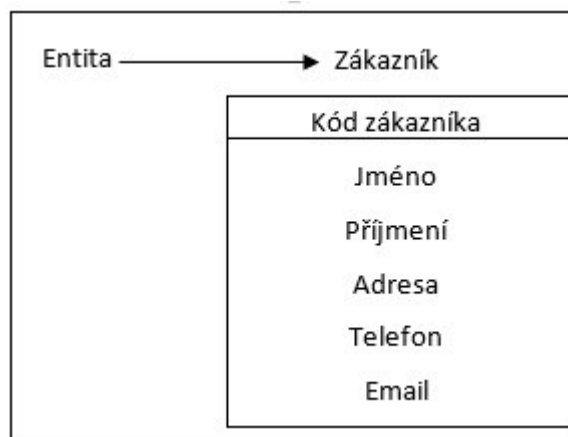
2.4.2 Konceptuální model

V konceptuálním modelu databáze je vytvořen model dat postavených na datech používaných v konkrétních organizacích, bez přemýšlení o detailech jako jsou jiné modely, nebo jiných úvah o pozdější fyzické implementaci. Konceptuální model identifikuje klíčové entity a jejich vztahy, které popisují situace reálného světa a je třeba je přenést do databáze. Z konceptuálního modelu databáze vychází logický model. (Connoly, Begg a Holowczak, 2009)

2.4.2.1 Entita

Entita je osoba, věc, nebo místo o které jsou shromažďovány data. Stručně řečeno entity jsou „věci“ z reálného světa, které jsou pro nás zajímavé, takže o nich zjišťujeme určité údaje a zaznamenáváme je do databáze. Při vytváření databáze samozřejmě entity vybíráme podle toho, s čím musí pro správnou funkci pracovat uživatelé. Každá entita, která se nachází v konceptuálním modelu, reprezentuje celou třídu entity. Jednotlivé výskyty entit označujeme jako její instance.

Na obrázku 2-4 můžeme vidět konkrétní entitu Zákazník, a její instance by mohl být například Petr Novák. Entita se značí obdelníkem. (Oppel, 2006)

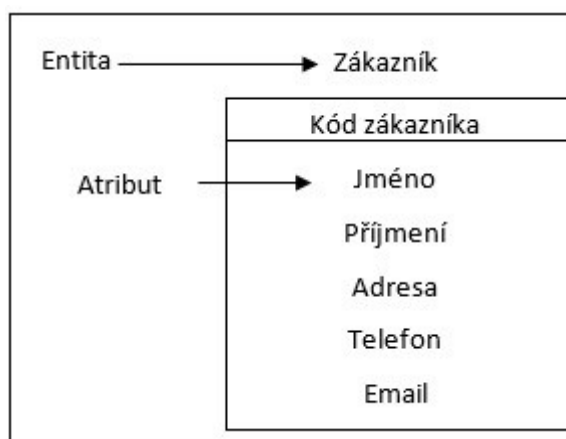


Obrázek 2-4 Ukázka entity, zdroj: Oppel (2006, s. 40), upraveno

2.4.2.2 Atribut

Atribut je jednotka faktu, vlastnost, která entitu nějakým způsobem charakterizuje. V konceptuálním modelu jsou znázorněny jako názvy uvnitř obdélníků (entity). Atributy uvedené v obdélníku nahoře značí jednoznačný identifikátor entity, který se nesmí opakovat. Každý atribut by měl být atomický, to znamená, že by nemělo jít jeho hodnotu dále rozdělit. Dále může být atribut jednoduchý nebo složený, jinými slovy se skládá z jednoho nebo více atributů. (Oppel, 2006)

Na obrázku 2-5 je vidět, jak vypadají typické atributy entity. Atribut Kód Zákazníka, který se nachází nad čarou, je jednoznačný a to znamená, že každý zákazník ho musí mít jiný. Atribut Adresa je podezřelý, jelikož se dá rozložit na další údaje a není atomický. Záleží na situaci, jak se bude atribut využívat a proto se doporučuje úzká spolupráce s budoucími uživateli. (Oppel, 2006)



Obrázek 2-5 Ukázka atributu, zdroj: Oppel (2006, s. 40), upraveno

2.4.2.3 Vztahy

Vztahy popisují spojení neboli asociace mezi entitami. Vztahy slouží k propojení dat, které spolu navzájem souvisí, ale jsou uloženy v rozdílných entitách. Při návrhu konceptuálního modelu je velice důležité správně pochopit jednotlivé vztahy mezi entitami, například kolik zaměstnanců se stará o jednu objednávku. Vztahy jsou klasifikovány do tří základních skupin, a to vztahy 1:1, 1:N a M:N. (Oppel, 2006)

Vztah 1:1

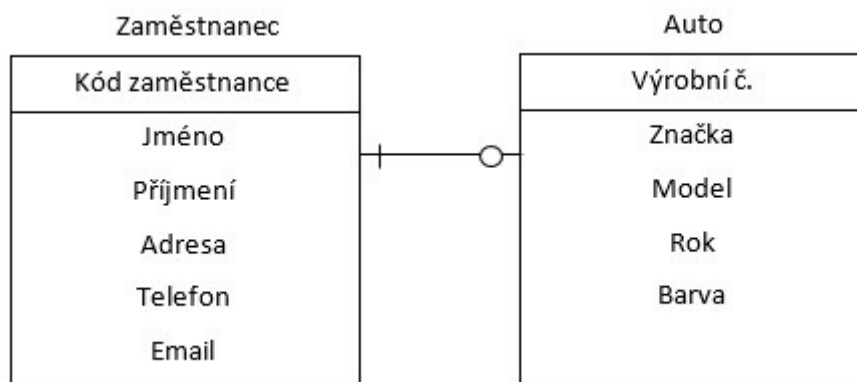
Vztah 1:1 je vztah, ve kterém je možné instanci jedné entity přiřadit maximálně k jedné instanci druhé entity. Tento vztah se v praxi moc nepoužívá, protože

symbolizuje chybu návrhu z toho důvodu, že obě entity lze nahradit snadno jednou. (Oppel, 2006)

Použití v praxi je možné, když:

- Při řešení otázek bezpečnosti u některých záznamů (instancí entit) je lze rozdělit do více tabulek
- Z důvodu přehlednosti tabulek, protože může nastat situace, kdy bude počet sloupců obrovský a v tabulce se nepůjde vyznat. (Kruczek, 2010)

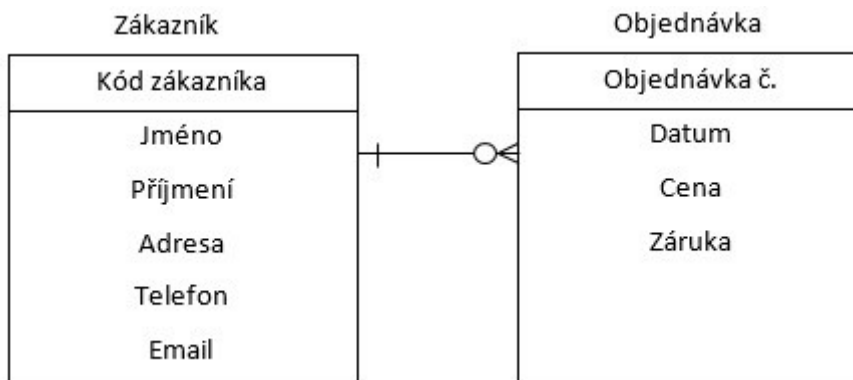
Na obrázku 2-6 je zobrazen vztah 1:1, který ukazuje, že jeden zaměstnanec má jedno služební auto.



Obrázek 2-6 Ukázka vztahu 1:1, zdroj: vlastní

Vztah 1:N

Vztahy typu 1:N jsou v databázích nejčastěji používané. Jedná se o vztah, kdy jakákoliv instance první entity může mít přiřazenou jednu nebo libovolný počet instancí druhé entity. Příkladem může být databáze zákazníků, kdy jeden zákazník má libovolný počet objednávek, ale jedna objednávka má jednoho zákazníka viz obrázek 2-7. (Oppel, 2006)

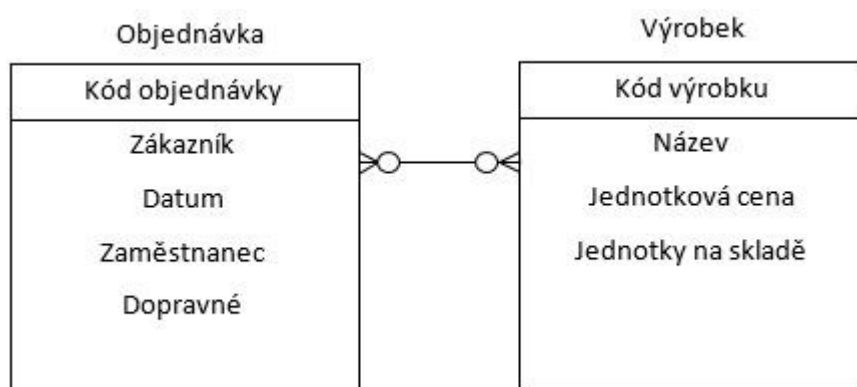


Obrázek 2-7 Ukázka vztahu 1:N, zdroj: vlastní

Vztah M:N

Vztah M:N je vztah, kdy nula až libovolný počet instancí první entity může být spojeno s nulou až libovolným počtem instancí druhé entity. V praxi jsou tyto vztahy běžné. (Oppel, 2006)

Na obrázku 2-8 je vidět vztah M:N. V jakémkoliv okamžiku může každá instance entity objednávka obsahovat libovolný počet výrobků a zároveň každá instance entity výrobek se může objevovat na libovolném počtu objednávek. (Oppel, 2006)



Obrázek 2-8 Ukázka vztahu M:N, zdroj: vlastní

V konceptuálním modelu jde tento typ vztahu bez problému zobrazit, ale když se přechází k fyzickému řešení tak nastává problém, protože relační model tento typ vztahu nepodporuje. Proto musí být vytvořena třetí pomocná tabulka, kde je použit vztah 1:N a spojit s ní dvě tabulky předchozí. (Oppel, 2006)

2.4.2.4 Klíče

Každá instance entity (záznam) v entitě (tabulce) musí být jednoznačně identifikovatelný, a proto je nutné určit atributy (sloupce), nebo kombinaci atributů (sloupců), které nám tuto jedinečnost dokážou zajistit. Těmto atributům se říká klíče. (Connolly, Begg a Holowczak, 2009)

Superklíč

Superklíč je: „*Atribut nebo množina atributů, která jedinečně identifikuje každý výskyt entity.*“ (Connolly, Begg a Holowczak, 2009, s. 161)

Superklíč může obsahovat sloupce, které nejsou pro identifikaci instance entity (záznamu) nutné, to znamená, že jsou nadbytečné a proto je třeba se zaměřit na klíče, které obsahují minimální počet atributů nutných pro jednoznačnou identifikaci instance

entity (záznamu). Takovým klíčům se říká kandidátní klíče. (Connoly, Begg a Holowczak, 2009)

Kandidátní klíč

Kandidátní klíč je: „*Superklíč, který obsahuje jen minimální počet atributů nezbytných k jedinečné identifikaci každého výskytu entit.*“ (Connoly, Begg a Holowczak, 2009, s. 161)

Pro kandidátní klíč platí:

- Podmínka jedinečnosti – Kandidátní klíč platí jen pro danou instanci entity (záznam), to znamená, že je pro každý záznam jedinečný a nemůžou existovat dva stejné kandidátní klíče.
- Podmínka neredukovatelnosti – Žádná podmnožina kandidátního klíče nezajišťuje jedinečnost, to znamená, že jestli se vynechá jakákoliv část kandidátního klíče, není zaručeno, že záznam bude jedinečný.
- Podmínka null – Kandidátní klíč nesmí obsahovat hodnotu null. (Connoly, Begg a Holowczak, 2009)

Primární klíč

Primární klíč je: „*Kandidátní klíč, který zvolíme pro jedinečnou identifikaci příslušné entity.*“ (Connoly, Begg a Holowczak, 2009, s. 161)

Na obrázku 2-9 je vidět ukázka primárního klíče jako atribut kód zákazníka u entity zákazník.

Zákazník	
PK Kód zákazníka	
Jméno	
Příjmení	
Adresa	
Telefon	
Email	

Obrázek 2-9 Ukázka Primárního klíče, zdroj: vlastní

Alternativní klíč

Alternativní klíč je: „*Kandidátní klíč, který není zvolen jako primární klíč příslušné entity.*“ (Connoly, Begg a Holowczak, 2009, s. 161)

2.4.3 Logický model

Logický návrh vychází z konceptuálního modelu, a v této práci používá relační databázový model jako svůj základ pro logické zobrazení databáze. V logickém modelu se převádí entity a jejich vztahy do relační podoby, to znamená do relačních tabulek. Z logického návrhu vychází návrh fyzický, který je podklad pro konkrétní programátory na fyzické úrovni a poskytuje informace a postupy pro fyzickou implementaci. (Connoly, Begg a Holowczak, 2009)

2.4.3.1 Tabulka

Tabulka je základ relačních databází a jeví se jako dvourozměrná struktura, kde sloupce reprezentují jednotlivé atributy a řádky jednotlivé záznamy. V logickém modelu jsou entity z konceptuálního modelu převedeny do tabulek. Většinou jednu entitu reprezentuje jedna tabulka, ale někdy může několik tabulek reprezentovat jednu entitu a naopak jedna tabulka může reprezentovat několik entit. (Oppel, 2006)

Tabulky v relačních databázích mají následující vlastnosti:

- Tabulka má jedinečný název v celé databázi, který ji jednoznačně identifikuje.
 - Všechny sloupce (atributy) jsou v rámci dané tabulky jednoznačně a jedinečně pojmenované.
 - Pořadí jednotlivých sloupců nehraje roli, to znamená, že pokud je sloupec na prvním nebo posledním místě tabulky ho nijak neovlivní, takže se můžou libovolně prohazovat.
 - V tabulce neexistují duplicitní záznamy, což zajišťuje správné zavedení primárního klíče.
 - Pořadí jednotlivých záznamu nehraje roli, to znamená, že pokud je záznam na prvním nebo posledním místě tabulky ho nijak neovlivní.
 - Hodnoty v jednotlivých buňkách by měly obsahovat jednu hodnotu.
- (Connoly, Begg a Holowczak, 2009)

2.4.3.2 Normalizace

V roce 1972 vyvinul E. F. Codd proces normalizace relačních databází, jehož cílem je aby jednotlivé relace (tabulky) splňovaly určitá pravidla a vytvořil tři normální formy, které byly později rozšířeny o další. (Oppel, 2006)

Normalizace je proces, kdy velké tabulky rozdělujeme na menší za účelem odstranění redundantních (nadbytečných dat) a zabránění případným problémům při následné aktualizaci tabulek. (Oppel, 2006)

Další skutečností je, že pokud tabulky nejsou normalizované, přinášejí nám problémy v momentě, když je potřeba je určitým způsobem aktualizovat. Tyto problémy se nazývají anomálie a dělí se na anomálie při vkládání, anomálie při odstraňování a anomálie při aktualizace. (Connoly, Begg a Holowczak, 2009; Oppel, 2006)

První normální forma (1NF)

Tabulka je v první normální formě, pokud některý z jejích atributů neobsahuje více hodnot. To znamená, že žádná buňka v tabulce nesmí obsahovat více hodnot, má svůj primární klíč a žádnou hodnotu nelze dále rozložit. (Oppel, 2006)

Tato normální forma je podstatná pro vytvoření správně fungujících relačních tabulek. Je tedy nezbytné mít tabulky alespoň v této formě, ale pro praxi se doporučuje normalizovat do třetí normální formy, aby bylo zabráněno problémům s anomáliemi. (Connoly, Begg a Holowczak, 2009)

Na obrázku 2-10 je vidět rozdíl mezi tabulkou, která neprošla procesem normalizace do 1NF a tabulkou, která jím prošla. Tabulka v 1NF obsahuje primární klíč a žádná hodnota v sloupci se neopakuje a nejde dále rozdělit.

tblZakaznik		tblZakaznik			
Jméno	Telefon	<u>Kód zákazníka</u>	Jméno	Příjmení	Telefon
Jan Peter	558623788	<u>JP1</u>	Jan	Peter	558623788
Miloš Kaša	558623350	<u>MK2</u>	Miloš	Kaša	558623350

ONF – nenormalizovaná tabulka

1NF – normalizovaná tabulka

Obrázek 2-10 Ukázka normalizace, zdroj: vlastní

Druhá normální forma (2NF)

Druhá normální forma souvisí jen s tabulkami, které mají složený primární klíč. Pokud má tabulka v první normální formě primární klíč z jednoho sloupce (atributu), je automaticky považována za druhou normální formu. Druhá normální forma tedy souvisí jen s tabulkami, které mají primární klíč složený ze dvou nebo více sloupců (atributů). (Connoly, Begg a Holowczak, 2009)

Funkční závislost nám vysvětluje vztah mezi sloupci v tabulce a primárním klíčem a ukazuje, že spolu sloupce navzájem souvisí. V druhé normální formě je potřeba, aby všechny sloupce byly plně závislé na primárním klíči. Při složeném primárním klíči můžou existovat sloupce, které jsou částečně závislé jen na části primárního klíče a můžou způsobovat problémy se vkládáním nových záznamů do databáze, jejich aktualizací nebo při odstraňování. Takle situace se musí řešit rozkladem tabulky na více tabulek. (Connoly, Begg a Holowczak, 2009)

Třetí normální forma (3NF)

Tabulka je ve třetí normální formě, pokud všechny neklíčové sloupce (atributy), jsou určeny pouze primárním klíčem. To znamená, že tabulka je ve třetí normální formě, pokud mezi neklíčovými sloupci neexistují tranzitivní závislosti a tabulka vychází z druhé normální formě. (Connoly, Begg a Holowczak, 2009)

Boyce – Coddova normální forma (BCNF)

Boyce – Coddova normální forma je vylepšenou variantou třetí normální formy. Jedná se o tabulky, kde se nachází více kandidátních klíčů a všechny musí být poskládány z více atributů. Tato normální forma říká, že žádný z atributů, který nezvolíme primárním klíčem, nesmí jednoznačně určovat hodnotu jiného atributu a mezi kandidátními klíči nesmí existovat funkční závislost. (Oppel, 2006)

Čtvrtá normální forma (4NF) a pátá normální forma (5NF)

Existuje i čtvrtá a pátá normální forma, ale v praxi se nepoužívají.

2.4.3.3 Omezení

Omezení je pravidlo, které v databázi nastavuje ve většině případů administrátor. Jedná se o omezení databázových objektů a ve většině případů konkrétně tabulek. Toto omezení definuje hodnoty, které lze do tabulek zadat. Na

úrovni logického modelování tyto omezení pomáhají vytvářet vztahy, na které je možné se odkazovat. (Oppel, 2006)

Omezení primárního klíče

Omezení primárního klíče znamená, že určitý sloupec, který je nastavený jako primární klíč zajišťuje jedinečnost každého záznamu a zamítá veškerou duplicitu. Pokud je primární klíč nastaven jako složený z více sloupců, jednotlivé sloupce jedinečné být nemusí, ale jejich kombinace musí vytvořit jedinečně identifikovatelný záznam. (Oppel, 2006)

Referenční omezení

Referenční omezení, jinými slovy referenční integrita kontroluje a zajišťuje vztahy mezi tabulkami takovým způsobem, že pokud je v příbuzné tabulce cizí klíč, tak musí mít odpovídající hodnotu primárního klíče v tabulce, kde je definován. (Oppel, 2006)

Omezení integrity

Omezení integrity je takové omezení, které pomáhá zvýšit kvalitu dat. Jinými slovy se nejčastěji jedná o omezení jednotlivých sloupců v rámci tabulek, mezi které patří:

- omezení NOT NULL - Hodnota musí být zadána, nesmí být nulová.
- omezení CHECK - Jedná se o ověřovací pravidlo, které nám vrací hodnotu pravda nebo nepravda, s tím že pokud záznam vrátí pravdu, můžeme ho do tabulky zadat. (Oppel, 2006)

2.4.4 Fyzický model

Fyzický model vychází z logického modelu a jedná se o implementaci logického modelu do konkrétního systému řízení báze dat (DBMS). Tento proces je náročný, a je zapotřebí zkušeného návrháře s detailní znalostí daného DBMS, protože pro něj musí návrh přizpůsobit. DBMS mohou být například Microsoft Access, Oracle, SQL Server, DB2. (Connolly, Begg a Holowczak, 2009)

2.5 SQL

SQL (Structured Query Language) je strukturovaný jazyk, který se používá pro práci s databázemi. V současné době je tento jazyk jasnou jedničkou pro relační databáze a pomocí něho lze vytvářet struktury jednotlivých databází a tabulek, provádět pravidelnou údržbu dat a jejich modifikaci (vkládání, úprava a mazání) a provádět jednoduché i složité dotazy. Mezi základní vlastnosti jazyka SQL patří:

- Jednoduchost – rychlé pochopení tohoto jazyka.
- Neprocedurální jazyk – říká databázi, co za výsledek požaduje, nikoliv jak toho má databáze dosáhnout.
- Volnost – Syntaxe není tak přísná a nemusíme dodržovat naprosto přesně stanovené pořadí některých znaků.
- Rozšířenost – Tento jazyk získal jako jediný vřelé uvítání od uživatelů a proto téměř všichni výrobci poskytují své produkty, podporované jazykem SQL. (Oppel, 2008; Connolly, Begg a Holowczak, 2009)

2.5.1 Kategorie příkazů

Příkazy jazyka SQL se dělí do čtyř kategorií podle toho, jakou funkci vykonávají. Pro všechny kategorie platí, že mají totožnou syntaxi a dodržují se u nich stejné pravidla. Tyto kategorie se nazývají DDL, DQL, DML a DCL a jsou popsány níže. (Oppel, 2008)

2.5.1.1 DDL

Jazyk DDL (Data Definition Language) slouží pro vytváření databázových objektů, jejich mazání nebo upravování. Hlavní příkazy, které jazyk DDL obsahuje, jsou:

- CREATE – slouží k vytvoření databázových objektů,
- ALTER – slouží k úpravě databázových objektů,
- DROP – slouží k odstraňování databázových objektů. (Oppel, 2008)

2.5.1.2 DML

Jazyk DML (Data Manipulation Language) slouží pro manipulaci s daty, jako je získávání dat z databází, vkládání nových záznamů, upravování existujících záznamů a odstraňování záznamů. Hlavní příkazy, které jazyk DML obsahuje, jsou:

- SELECT – slouží pro výběr dat z databáze,
- INSERT – slouží pro vkládání nových dat (záznamů),
- UPDATE – slouží pro úpravu stávajících dat (záznamů),
- DELETE – slouží pro odstraňování dat (záznamů). (Oppel, 2008)

2.5.1.3 DCL

Jazyk DCL (Data Control Language) slouží pro správu dat, a zajištění bezpečnosti v rámci databáze. Tento jazyk používají administrátoři databází a pomocí něho přidělují různá oprávnění v rámci konkrétního DBMS. Mezi některé příkazy například patří:

- CREATE USER – slouží k vytvoření uživatele,
- GRANT – přidělí v rámci databáze oprávnění,
- DROP USER – slouží k odstranění uživatele. (Oppel, 2008)

2.5.1.4 TCC

Jazyk TCC (Transaction Control Commands) slouží pro řízení databázové transakce. Databázová transakce je uspořádaná množina příkazů, která se jeví jako ucelená jednotka. Transakce se musí celá provést úspěšně, jinak se databáze vrátí do původního stavu. Mezi některé příkazy pro řízení transakcí například patří:

- BEGIN TRANSACTION – slouží k nastavení požadovaného bloku na transakci,
- COMMIT – slouží k potvrzení transakce,
- ROLLBACK – při chybě vrátí databázi do původního stavu. (Oppel, 2008)

2.6 Microsoft Access

Microsoft Access je aplikace, která pracuje s databázemi. Funguje jako nástroj pro správu relačních databází a patří do balíčku Microsoft Office a proto disponuje velmi podobným uživatelským rozhraním pro ovládání jako jiné programy v tomto balíčku. Prostřednictvím databázových ovladačů přistupuje k téměř jakékoliv databázi, například k Microsoft SQL Server, Oracle a mnoho dalším. Tato aplikace nabízí možnosti, mezi které například patří:

- Ukládat data.
- Manipulace s daty – můžeme si data prohlížet, upravovat, přidávat nové a editovat.

- Poskytovat data – k datům může přistupovat například tiskárna, obrazovka
- Importovat data z jiných zdrojů, databázi, ale také je můžeme exportovat.
- Možnost sdílení (Kruczek, 2010; Novák, 2014)

Mezi základní součásti aplikace Microsoft Access patří tabulka, dotazy, formuláře a sestavy, které popíšeme níže. (Kruczek, 2010)

2.6.1 Tabulky

Pojem tabulka už byl v této práci vysvětlen, ale pro zopakování se jedná o dvourozměrnou strukturu, která obsahuje sloupce (atributy) a řádky (záznamy). (Kruczek, 2010)

2.6.1.1 Vytvoření tabulky

Tabulky jsou v relačních databázích základem, a při práci s programem Microsoft Access je znalost jejich vytváření velice důležitá. Jednotlivé tabulky se mohou používat z různých důvodů, jejich obsah se může lišit a jejich využívání může probíhat na více úrovních. Microsoft Access nabízí tyto základní možnosti pro vytváření tabulek:

- Prázdná tabulka – k tomuto vytvoření využijeme tlačítko Tabulka.
- Tabulka v návrhovém zobrazení – k tomuto vytvoření využijeme tlačítko Návrh Tabulky a jedná se o nejpoužívanější možnost, kde můžeme definovat jednotlivé sloupce a datové typy.
- Tabulka propojená se službou SharePoint – slouží pro práci více uživatelů nad jednou tabulkou v prostředí Microsoft Access. (Kruczek, 2010)

2.6.1.2 Přehled datových typů

Datový typ v tabulce slouží k určení toho, jaké hodnoty bude moct uživatel do jednotlivých polí zadat. Mezi základní datové typy aplikace Microsoft Access patří:

- Text – slouží k zápisu textových hodnot do jednotlivých polí a jeho délka je omezena na 255 znaků
- Memo – slouží k zápisu dlouhých textových hodnot a využívá se hlavně při popisech a podrobnostech jednotlivých záznamů. Může obsahovat až 65 535 znaků.
- Číslo – jak už název napovídá, slouží k ukládání číselných údajů.

- Datum a čas – používá se pro ukládání data a času.
- Měna – Slouží k ukládání peněžních a finančních hodnot.
- Ano – Ne – Reprezentuje hodnoty pravda - nepravda
- Objekt OLE – Slouží pro vkládání dat, které databáze nedokáže jednoznačně interpretovat. Typickým příkladem jsou například obrázky nebo pdf soubory
- Hypertextový odkaz – Slouží k odkazování na jiné internetové adresy pomocí URL nebo na konkrétní fyzické soubory. (Kruczek, 2010)

2.6.1.3 Vlastnosti datových typů

V programu Microsoft Access disponuje každý datový typ určitými vlastnostmi, které určitým způsobem specifikují zadávání dat do tabulek. Mezi základní vlastnosti patří:

- Velikost Pole – určuje velikost jednotlivých polí a přiděluje jim patřičnou paměťovou kapacitu. Pro text se určuje počet znaků, pro číslo přesnost.
- Formát – určuje způsob zobrazení jednotlivých polí. Například krátké datum jenom s uvedením dnů a měsíců nebo dlouhé datum, kde se uvádí i rok.
- Vstupní maska – jednoznačně specifikuje způsob, jakým se budou do polí zadávat hodnoty. Používá se například u PSČ nebo telefonních čísel.
- Titulek – je podrobnější popis, který se zobrazuje jako název jednotlivých sloupců v datovém listu.
- Ověřovací pravidlo – určuje kritérium, které musí být splněno, jinak není možné zadat údaj do pole.
- Je nutno zadat – při vložení každého nového záznamu musí být toto pole vyplněno.
- Indexovat – používá se při častém vyhledávání, nebo filtraci pomocí těchto polí a urychluje tak chod databáze. Je možno indexovat s duplicitou nebo bez duplicity. (Kruczek, 2010)

2.6.2 Dotazy

V jednotlivých tabulkách jsou uložena data, ale je třeba k nim nějakým způsobem přistupovat, pracovat s nimi, vytvářet nové a mnoho dalšího. K tomu všemu slouží dotazy. Mezi základní vlastnosti a výhody dotazů patří následující:

- Dotaz se vytváří jen jednou a poté ho lze opakovaně používat.

- Dotaz disponuje velkým množstvím možností, jak lze s jednotlivými daty pracovat.
- Dotaz nabízí funkci parametru, to znamená, že při každém použití bude vyžadovat parametr, podle které se vykoná
- Dotaz umožňuje funkci souhrnů, mezi které patří například sčítání hodnot jednotlivých záznamů, výpočet průměru a mnoho dalšího.
- Microsoft Access nabízí uživatelské rozhraní pro tvorbu dotazů, které pak převádí do SQL jazyka, ale možnost psát dotazy přímo v SQL nám zůstává. (Kruczek, 2010)

Microsoft Access nabízí tvorbu čtyř skupin dotazů, které jsou:

- Výběrové dotazy – využívá se k zobrazení konkrétních dat z jednotlivých tabulek.
- Křížové dotazy – speciální typ dotazu, který se používá často pro souhrnné funkce jako je například součet a vytváří speciální záhlaví.
- Akční dotazy – určitým způsobem vytvářejí, odstraňují, aktualizují nebo přidávají jednotlivá data.
- Dotazy SQL – Sjednocovací, předávací a definiční. (Kruczek, 2010)

2.6.3 Formuláře

Formuláře slouží jako rozhraní pro práci s databází. Umožňují jednoduše zadávat data do jednotlivých tabulek a taky si je prohlížet. Uživatel tak může v jednom formuláři nahlížet na data klidně ze čtyř tabulek, které spolu souvisí a patřičně s nimi manipulovat. (Kruczek, 2010)

Formuláře vypadají jako určitá okna, díky kterým uživatel komunikuje s databází. Zajišťují nám také vyšší bezpečnost a korektnost při vkládání záznamů do jednotlivých tabulek. (Kruczek, 2010)

2.6.3.1 Visual Basic For Application (VBA)

Je programovací jazyk od společnosti Microsoft a vychází ze svého předchůdce Visual Basic. Využívá se například v Microsoft Excel a hlavně v programu Microsoft Access. (Shepherd, 2012)

Jedná se o procedurální programovací jazyk, který je strukturovaný. Základním prvkem tohoto programu jsou moduly, do kterých se zapisují jednotlivé programy a funkce. V programu Microsoft Access se využívá tento jazyk hlavně při tvorbě formulářů. (Shepher, 2012; Novák, 2014)

Moduly jsou oddíly se zdrojovým kódem, které jsou součástí naší aplikace. V rámci těchto modulů je možno deklarovat procedury, které souvisí s jednotlivými prvky aplikace, jako jsou například formuláře. Tyto procedury je nutné přímo zavolat. Microsoft Access nabízí dva základní typy modulů a to jsou:

- Moduly třídy – jedná se o moduly formulářů a jednotlivých sestav.
- Moduly standartní – využívá se pro univerzální funkce a procedury. (Shepher, 2012; Novák, 2014)

2.6.4 Sestavy

V programu Microsoft Access sestavy reprezentují výstup. Ve většině případů se jedná o data, která jsou určena k tisku. V rámci sestav je možné je upravovat a vytvářet tak šablony, do kterých se pak jen doplní potřebná data z databáze. Typickým příkladem v obchodní sféře mohou být faktury, dodací listy nebo objednávky. Sestavy se svým vytvářením velmi podobají formulářům. (Kruczek, 2010)

3 Analýza současného stavu ve firmě

Cílem této kapitoly je seznámení s firmou, kde bude bakalářská práce realizována včetně analýzy současného stavu, jakým způsobem své data zpracovávají. Poslední části této kapitoly jsou požadavky na vytvoření nové databázové aplikace.

3.1 O společnosti

Firma EkoMedica Czech, s.r.o. obchoduje s produkty, které určitým způsobem obohacují lidský organismus. Nejčastěji se jedná o přírodní doplňky stravy a homeopatika. Jednotlivé produkty dovážejí od svých zahraničních dodavatelů a skladují je v České Republice ve svých pronajatých skladech. Následně je pak distribuují do různých lékáren nebo obchodů se zdravou výživou.

Jedná se o skupinu živnostníků, kteří vzájemně spolupracují a využívají stejné dodavatele a odběratele pro prodej pod stejnou značkou. Databázová aplikace bude vytvořena pro konkrétního živnostníka a jeho potřeby, kde bude testována její přínosnost. Pokud se tato aplikace v praxi osvědčí, bude rozšířena a integrovaná pro celou firmu EkoMedica Czech s.r.o.

3.2 Současná situace

V současné době není využíván žádný systém, který by určitým způsobem zjednodušil a zefektivnil práci s jednotlivými objednávkami a dodávkami. Všechna tato činnost je prováděna pomocí programu Microsoft Excel. Vyřízení jedné větší objednávky tímto způsobem je enormně časově náročné a značně neefektivní.

Začíná docházet k velmi nepříjemným nedorozuměním se zákazníky i dodavateli, jelikož s každou objednávkou, dodávkou, novým produktem a mnohým dalším přibývají nová data, která už nejsou schopni pomocí programu Microsoft Excel efektivně zpracovávat. Objevují se tak špatně vyplněné dodací listy, objednávky a dochází k velkým časovým prodlevám, pokud je třeba například řešit reklamace.

3.3 Požadavky

Požadavky byly analyzovány na základě několika schůzek s konkrétním živnostníkem v dané firmě.

Jeden z prvních požadavků je důraz na přehlednost, jednoduchost a intuitivní ovládání celé aplikace, jelikož uživatel disponuje jen základními znalostmi informačních technologií.

Hlavním požadavkem je vytvořit databázovou aplikaci, která bude ulehčovat vyřizování jednotlivých objednávek a dodávek a jejich následnou archivaci na jednom místě a na základě toho vytvářet jednotlivé dodací a objednávkové listy.

Dále je nutné uchovávat informace o jednotlivých dodavatelích a odběratelích s možností je určitým způsobem modifikovat. Pro každého dodavatele a odběratele by měly být uchovány základní informace, které jsou nutné pro obchodní činnost.

Posledním požadavkem je zaznamenávání údajů o jednotlivých produktech s možností přidávat nové a editovat stávající. Ke každému produktu by měl být uveden také popis, který bude využit pro tisk seznamu produktů pro jednotlivé odběratele.

Budoucí uživatel si od databázové aplikace slibuje výrazné zefektivnění jeho dosavadní činnosti při evidenci objednávek a dodávek a tím i velkou časovou úsporu. Dále bude mít k dispozici na jednom místě všechny data, a nebude muset dohledávat jednotlivé soubory v programu Microsoft Excel, jak tomu bylo doposud. Z toho vyplývá zrychlení vyřizování případných reklamací, stížností a nedorozumění.

4 Návrh a implementace databázové aplikace

V první části této kapitoly bude databáze navržena do konceptuálního modelu, který zobrazuje určité zobecnění nad skutečnou databázovou strukturou. V rámci tohoto modelu budou popsány jednotlivé entity, atributy a vztahy, které byly identifikovány jako klíčové pro návrh aplikace. Následně bude konceptuální model převeden do logického modelu, konkrétně relačního modelu, což znamená, že jednotlivé entity budou převedeny do tabulek s primárními a cizími klíči, a jednotlivé sloupce budou obsahovat datové typy. Nakonec bude logický model převeden do fyzického modelu, což znamená do konkrétní implementace v programu Microsoft Access.

Druhá část této kapitoly bude věnována návrhu jednotlivých tabulek v programu Microsoft Access, kde budou popsány všechny atributy jednotlivých tabulek včetně primárních klíčů a vysvětlen význam všech tabulek. Po vytvoření těchto tabulek do nich budou nalita zkušební data, která budou testovat správnou funkci po dobu návrhu celé aplikace.

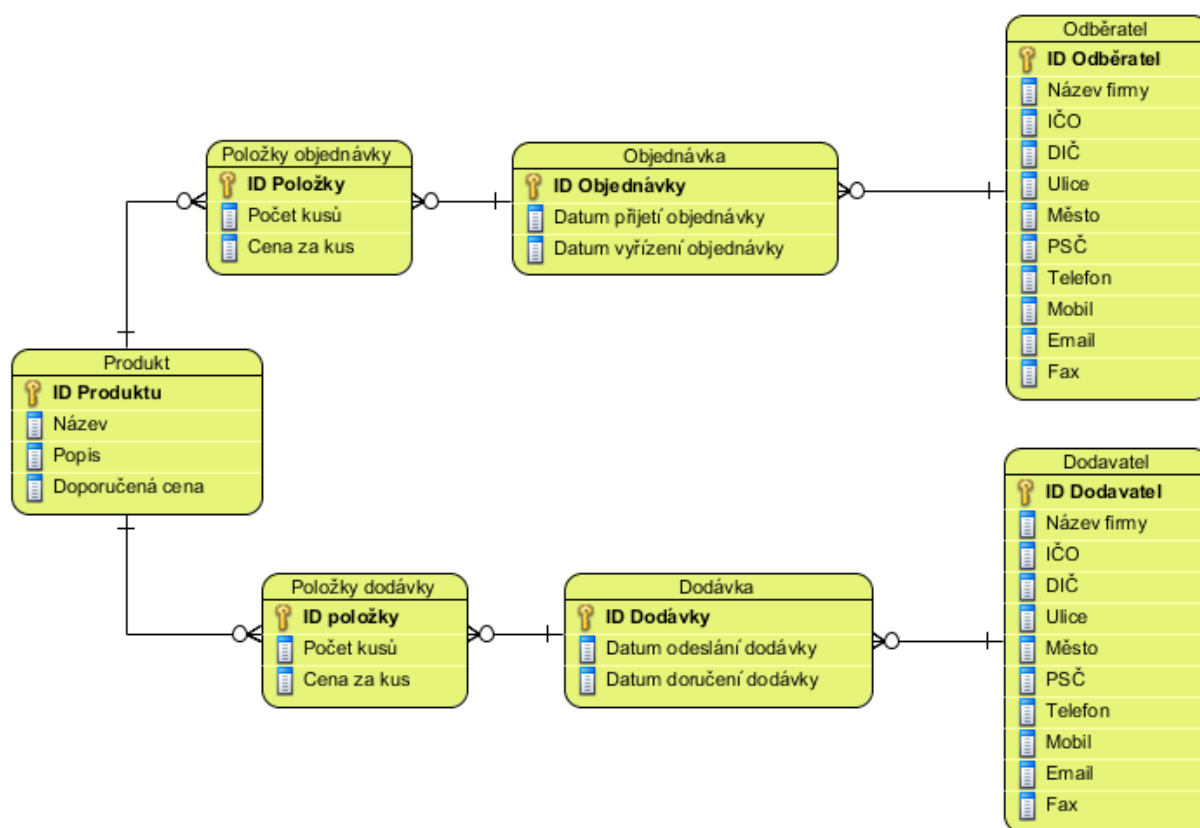
Třetí část této kapitoly bude zaměřena na konkrétní aplikační funkce, které budou zajišťovat jednotlivé formuláře a sestavy. Na základě požadavků zadavatele, bude vytvořen jednoduchý grafický design, který bude připomínat lékařské prostředí. Funkčnost jednotlivých formulářů bude zajišťována pomocí programovacího jazyka VBA. Bude zajištěna podpora tvorby dodacího a objednávkového listu pomocí sestav. Komunikace formulářů bude probíhat ve většině případů v rámci dialogů.

V poslední části bude tato aplikace převedena uživatelského prostředí, kde nebude mít běžný uživatel přístup k jednotlivým databázovým položkám a aplikace bude uložena na určitý typ přenosového média, například CD.

4.1 Konceptuální návrh databáze

Konceptuální návrh databáze vycházel z analýzy požadavků zadavatele a byly identifikovány základní entity včetně jejich atributů a vzájemných vztahů, které jsou nutné pro tvorbu efektivní databázové aplikace.

Na obrázku 4-1 je znázorněn konceptuální model, ve kterém jsou identifikovány všechny důležité entity a jejich atributy, propojené vzájemnými vztahy. V horní části jednotlivých entit se nachází jejich název a v jejich těle potom atributy, kde značka klíče značí jednoznačný identifikátor, čili primární klíč. Všechny entity jsou propojeny vztahy 1:N.

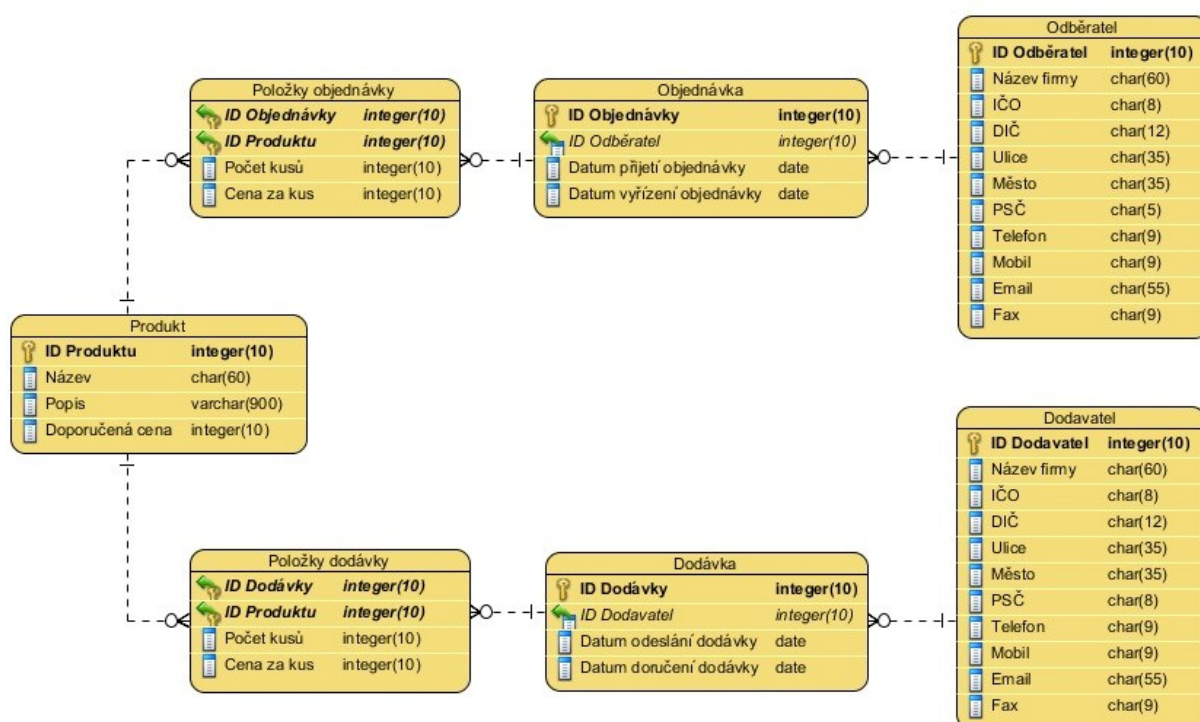


Obrázek 4-1 Konceptuální model databáze, zdroj: vlastní

4.2 Logický návrh databáze

Při návrhu logického modelu databáze bylo vycházeno z konceptuálního modelu. Jednotlivé entity byly převedeny do podoby tabulek, a jednotlivé atributy do podoby sloupců, které obsahují blíže specifikované datové typy. Pro správné propojení byly přidány do určitých tabulek cizí klíče.

Na obrázku 4-2 je znázorněn logický model databázové aplikace, který už obsahuje jednotlivé tabulky. Do tabulky *Objednávka* a *Dodávka* byly přidány cizí klíče, které specifikují vztah s tabulkou *Odběratel* nebo tabulkou *Dodavatel*. Z důvodu vytváření dodacích a objednávkových listů podle jasně stanovené předlohy byl u tabulek *Položky objednávky* a *Položky dodávky* nahrazen primární klíč *ID položky* složeným primárním klíčem ze sloupců *ID Objednávky* a *ID Produktu* nebo *ID Dodávky* a *ID Produktu*. Tyto zároveň cizí klíče specifikují vztah mezi tabulkami.

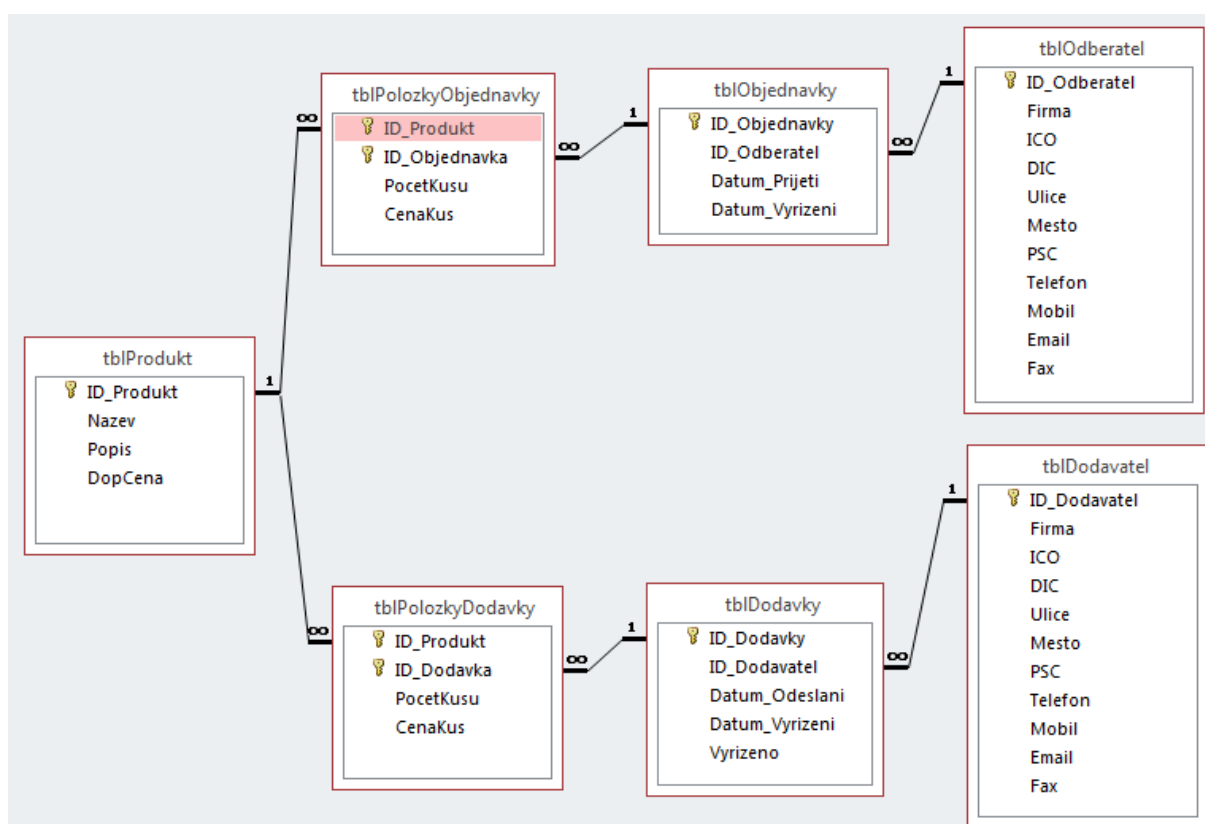


Obrázek 4-2 Logický model databáze, zdroj: vlastní

4.3 Fyzický návrh databáze

Fyzický návrh databáze vychází z logického návrhu databáze, kde se jedná o implementaci pro konkrétní *systém řízení báze dat*.

Na obrázku 4-3 je znázorněn fyzický model databáze, který je přizpůsobený pro konkrétní *SŘBD* programu *Microsoft Access*. Na tomto obrázku jsou znázorněny finální podoby jednotlivých tabulek pro databázovou aplikaci. Propojení je realizováno pomocí primárních a cizích klíčů, kdy například tabulka *tblOdberatel* je propojena s tabulkou *tblObjednavky* prostřednictvím cizího klíče *ID_Odberatel*, který je umístěný v tabulce *tblObjednavky*.



Obrázek 4-3 Fyzický model databáze, zdroj: vlastní

4.4 Návrh tabulek

Důležitou součástí správného návrhu databázové aplikace v programu Microsoft Access je správné navržení jednotlivých tabulek. V této kapitole jsou popsány jednotlivé tabulky včetně jejich významu. Pro každou tabulku je vytvořen přehled pro jednotlivé sloupce, kde jsou popsány jejich datové typy včetně vybraných vlastností.

4.4.1 Tabulka tblOdberatel

Tabulka tblOdberatel slouží k ukládání údajů o zákaznících, kterým jsou produkty distribuovány. Nejčastěji se jedná o lékárny a obchody se zdravou výživou.

Tabulka obsahuje jedenáct sloupců, kde sloupec *ID_Odberatel* představuje primární klíč, jehož datový typ je automatické číslo, které se vždy vygeneruje při přidání nového odběratele. Všechny ostatní sloupce (pole) jsou datového typu text a v závorce jsou uvedeny jejich jednotlivé velikosti. Patří mezi ně sloupec *Firma* (60), který reprezentuje název odběratele (zákazníka). Sloupec *ICO* (8), který slouží k uložení identifikačního čísla osob. U tohoto sloupce je nastavena vstupní maska pro zadání hodnot. Sloupec *DIC* (12) slouží k uložení daňového identifikačního čísla a má také nastavenou vstupní masku. Sloupce *Ulice* (35), *Mesto* (35) a *PSC* (5) u kterého je nastavena vstupní maska, slouží k uložení adresy. Posledními sloupci v této tabulce jsou *Telefon* (9), *Mobil* (9), *Email* (55), *Fax* (9) a slouží k ukládání kontaktních informací. U sloupců *Telefon* a *Mobil* je nastavena vstupní maska. V tabulce 4-1 je uvedeno podrobné zobrazení jednotlivých sloupců (polí) včetně jejich datových typů a základních vlastností.

Tabulka 4-1 Návrh tabulky tblOdberatel, zdroj: vlastní

Název pole	Datový typ	Velikost	Vstupní maska	Nutno zadat	Indexovat
ID_Odberatel	Automatické č.	Dlouhé celé č.	-	-	Bez duplicity
Firma	Text	60	-	Ano	Ne
ICO	Text	8	00000000	Ano	Ne
DIC	Text	12	LL0000000000	Ano	Ne
Ulice	Text	35	-	Ano	Ne
Mesto	Text	35	-	Ano	Ne
PSC	Text	5	000\ 00	Ano	Ne
Telefon	Text	9	"+558 "00\ 00\ 00	Ne	Ne
Mobil	Text	9	"+420 "000\ 000\ 000	Ne	Ne

Email	Text	55	-	Ne	Ne
Fax	Text	9	-	Ne	Ne

4.4.2 Tabulka tblDodavatel

Tabulka tblDodavatel slouží k ukládání údajů o dodavatelích, od kterých jsou produkty nakupovány. Ve většině případů se jedná o zahraniční výrobce těchto produktů.

Tabulka je svým návrhem téměř totožná s tabulkou *tblOdberatel* (tabulka 4-1) a liší se jen v primárním klíči, který reprezentuje sloupec *ID_Dodavatel*. Tento sloupec je datového typu automatické číslo a při přidání nového dodavatele se automaticky vygeneruje. V tabulce 4-2 je uvedeno podrobné zobrazení jednotlivých sloupců (polí) včetně jejich datových typů a základních vlastností.

Tabulka 4-2 Návrh tabulky tblDodavatel, zdroj: vlastní

Název pole	Datový typ	Velikost	Vstupní maska	Nutno zadat	Indexovat
ID_Dodavatel	Automatické č.	Dlouhé celé č.	-	-	Bez duplicity
Firma	Text	60	-	Ano	Ne
ICO	Text	8	00000000	Ano	Ne
DIC	Text	12	LL0000000000	Ano	Ne
Ulice	Text	35	-	Ano	Ne
Mesto	Text	35	-	Ano	Ne
PSC	Text	5	000\ 00	Ano	Ne
Telefon	Text	9	" +558 "00\ 00\ 00	Ne	Ne
Mobil	Text	9	" +420 "000\ 000\ 000	Ne	Ne
Email	Text	55	-	Ne	Ne
Fax	Text	9	-	Ne	Ne

4.4.3 Tabulka tblProdukt

Tabulka tblProdukt slouží k ukládání údajů o produktech, které má firma ve své nabídce a distribuuje je svým odběratelům.

Tabulka obsahuje čtyři sloupce, kde sloupec *ID_Produkt* slouží jako primární klíč a jeho datový typ je automatické číslo, které se generuje při přidání nového produktu. Dalším sloupcem je *Nazev* (100), který je datového typu text a číslo v závorce opět reprezentuje velikost. Jak už vychází z názvu, tento sloupec ukládá název produktu. Sloupec *Popis* je datového typu memo, který slouží k ukládání dlouhých textových řetězců. Do tohoto sloupce se ukládá podrobný popis produktu. Posledním sloupcem je *DopCena* a jeho datový typ je měna. Používá se pro ukládání doporučených cen k prodeji jednotlivých produktů za kus. V tabulce 4-3 je uvedeno podrobné zobrazení jednotlivých sloupců (polí) včetně jejich datových typů a základních vlastností.

Tabulka 4-3 Návrh tabulky tblProdukt, zdroj: vlastní

Název pole	Datový typ	Velikost	Vstupní maska	Nutno zadat	Indexovat
ID_Produkt	Automatické č.	Dlouhé celé č.	-	-	Bez duplicity
Nazev	Text	100	-	Ano	Ne
Popis	Memo	-	-	Ne	Ne
DopCena	Měna	Měna	-	Ne	Ne

4.4.4 Tabulka tblObjednavky

Tabulka tblObjednavky slouží k ukládání údajů o jednotlivých objednávkách odběratelů, respektive zákazníků.

Tabulka obsahuje čtyři sloupce, kde sloupec *ID_Objednavky* slouží jako primární klíč a jeho datový typ je opět automatické číslo, které se generuje při přidání nové objednávky. Další sloupec je *ID_Odberatel*, který funguje jako cizí klíč a slouží k vytvoření vztahu s tabulkou *tblOdberatel*. Datovým typem je číslo, které odpovídá hodnotě *ID_Odberatel* z tabulky *tblOdberatel*. Následující sloupec je *Datum_Prijeti*, který ukládá datum, kdy byla objednávka obdržena. Datovým typem tohoto sloupce je

datum a čas. Poslední sloupec je *Datum_Vyrizeni*, který slouží k uložení data, kdy byla objednávka vyřízena. Datový typ je opět datum a čas. V tabulce 4-4 je uvedeno podrobné zobrazení jednotlivých sloupců (polí) včetně jejich datových typů a základních vlastností.

Tabulka 4-4 Návrh tabulky *tblObjednavky*, zdroj: vlastní

Název pole	Datový typ	Velikost	Vstupní maska	Nutno zadat	Indexovat
ID_Objednavky	Automatické č.	Dlouhé celé č.	-	-	Bez duplicity
ID_Odberatel	Číslo	Dlouhé celé č.	-	Ano	Ne
Datum_Prijeti	Datum a čas	Datum	-	Ano	Ne
Datum_Vyrizeni	Datum a čas	Datum	-	Ne	Ne

4.4.5 Tabulka *tblDodavky*

Tabulka *tblDodavky* slouží k ukládání údajů o jednotlivých dodávkách od dodavatelů.

Tabulka obsahuje pět sloupců, kde sloupec *ID_Dodavky* slouží jako primární klíč a je opět datového typu automatické číslo, které se generuje při přidání nové dodávky. Dalším sloupcem je *ID_Dodavatel*, který funguje jako cizí klíč a slouží k vytvoření vztahu s tabulkou *tblDodavatel*. Datový typ tohoto sloupce je číslo, které odpovídá hodnotě primárního klíče *ID_Dodavatel* z tabulky *tblDodavatel*. Následující sloupec je *Datum_Odeslani*, který ukládá datum, kdy byla žádost o dodávku odeslána. Datovým typem je datum a čas. Sloupec *Datum_Vyrizeni* slouží k uložení data, kdy má být dodávka uskutečněna. Datovým typem je opět datum a čas. Posledním sloupcem v této tabulce je *Vyrizeno*, který je datového typu Ano / Ne a ukládá se do něj hodnota pravda, pokud již bylo zboží dodáno a nepravda, pokud ne. V tabulce 4-5 je uvedeno podrobné zobrazení jednotlivých sloupců (polí) včetně jejich datových typů a základních vlastností.

Tabulka 4-5 Návrh tabulky *tblDodavky*, zdroj: vlastní

Název pole	Datový typ	Velikost	Vstupní maska	Nutno zadat	Indexovat
ID_Dodavky	Automatické č.	Dlouhé celé č.	-	-	Bez duplicity
ID_Dodavatel	Číslo	Dlouhé celé č.	-	Ano	Ne
Datum_Odeslani	Datum a čas	Datum	-	Ano	Ne
Datum_Vyřízení	Datum a čas	Datum	-	Ano	Ne
Vyřizeno	Ano / Ne	-	-	-	-

4.4.6 Tabulka *tblPolozkyObjednavky*

Tabulka *tblPolozkyObjednavky* slouží k ukládání údajů o jednotlivých položkách pro konkrétní objednávky. Například objednávka může obsahovat několik různých produktů, jinými slovy položek.

Tabulka obsahuje čtyři sloupce, kde sloupce *ID_Produkt* a *ID_Objednavka* fungují jako složený primární klíč. Tento složený primární klíč je vytvořen na základě přesně stanovených předloh pro dodací list. Oba sloupce jsou datového typu číslo. Dále sloupec *ID_Produkt* funguje jako cizí klíč a vytváří vztah s tabulkou *tblProdukt*, kde hodnota odpovídá hodnotě primárního klíče *ID_Produkt* z tabulky *tblProdukt*. Jako cizí klíč funguje také sloupec *ID_Objednavka*, který vytváří vztah s tabulkou *tblObjednavky* a jeho hodnota odpovídá hodnotě primárního klíče *ID_Objednavky* z tabulky *tblObjednavky*. Následující sloupec *PocetKusu* slouží k uložení počtu kusů daného produktu (položky) a jeho datový typ je číslo. Poslední sloupec je *CenaKus*, do kterého se ukládá cena za jeden kus dané položky. Datový typ je opět číslo. V tabulce 4-6 je uvedeno podrobné zobrazení jednotlivých sloupců (polí) včetně jejich datových typů a základních vlastností.

Tabulka 4-6 Návrh tabulky *tblPolozkyObjednavky*, zdroj: vlastní

Název pole	Datový typ	Velikost	Vstupní maska	Nutno zadat	Indexovat
ID_Produkt	Číslo	Dlouhé celé č.	-	Ano	S duplicitou
ID_Objednavka	Číslo	Dlouhé celé č.	-	Ano	S duplicitou

PocetKusu	Číslo	Celé číslo	-	Ano	Ne
CenaKus	Měna	Měna	-	Ano	Ne

4.4.7 Tabulka tblPolozkyDodavky

Tabulka tblPolozkyDodavky slouží k ukládání údajů o jednotlivých položkách pro konkrétní dodávky.

Tato tabulka je téměř totožná jako tabulka *tblPolozkyObjednavky* a jediným rozdílem je nahrazení sloupce *ID_Objednavka* z tabulky *tblPolozkyObjednavky* sloupcem *ID_Dodavka*, který slouží jako cizí klíč a vytváří vztah s tabulkou *tblDodavky* a jeho hodnota odpovídá hodnotě primárního klíče *ID_Dodavky* z tabulky *tblDodavky*. V tabulce 4-7 je uvedeno podrobné zobrazení jednotlivých sloupců (polí) včetně jejich datových typů a základních vlastností.

Tabulka 4-7 Návrh tabulky tblPolozkyDodavky, zdroj: vlastní

Název pole	Datový typ	Velikost	Vstupní maska	Nutno zadat	Indexovat
ID_Produkt	Číslo	Dlouhé celé č.	-	Ano	S duplicitou
ID_Dodavka	Číslo	Dlouhé celé č.	-	Ano	S duplicitou
PocetKusu	Celé číslo	Dlouhé celé č.	-	Ano	Ne
CenaKus	Měna	Měna	-	Ano	Ne

4.5 Návrh formulářů a sestav

V této databázové aplikaci je vytvořen jeden hlavní navigační formulář, prostřednictvím kterého se přistupuje k jednotlivým dílčím formulářům. Grafický design aplikace je vytvořený na základě analýzy požadavku, kdy zadavatel požadoval rozhraní, které bude jednoduché, intuitivně ovladatelné a barevně bude charakteristické pro zdravý životní styl. Také je k dispozici tisk jednotlivých sestav, jako je seznam dodavatelů, odběratelů nebo produktů. Pro každou objednávku je možné vytvořit dodací list a pro každou dodávku objednávkový list.

4.5.1 Navigační formulář frmUvod

Na obrázku 4-4 je zobrazen úvodní formulář, který se zobrazí ihned po zapnutí aplikace. Slouží jako navigace pro jednotlivé dílčí formuláře a propojuje je v jeden celek. Tento formulář umožňuje přistupovat k informacím o jednotlivých produktech, dodavatelích, odběratelích, objednávkách, dodávkách a k tisku jednotlivých sestav. Obsahuje tlačítko *Konec*, pomocí něhož lze aplikaci vypnout.



Obrázek 4-4 Formulář frmUvod, zdroj: vlastní

4.5.2 Formulář frmProdukt

Pro přechod na formulář *frmProdukt* je k dispozici tlačítko *Produkt* na navigačním formuláři *frmUvod*. Prostřednictvím tohoto formuláře si lze jednotlivé produkty prohlížet, přidávat nové, editovat stávající nebo je vymazat. K dispozici je také možnost tisku.

Na obrázku 4-5 lze vidět konkrétní zobrazení formuláře *frmProdukt* pro produkt s názvem *Aloe*. V horní části formuláře lze přepínat mezi jednotlivými produkty prostřednictvím *pole se seznamem*. Jednotlivá *textová pole* jsou nepřístupná a pro editaci záznamů jsou k dispozici čtyři tlačítka, pomocí kterých může uživatel *Přidat produkt*, *Upravit*, *Odstranit* nebo *Vytisknout* detailní informace. V pravém dolním rohu obrazovky je tlačítko *Zavřít*, které slouží k navrácení na navigační formulář.

Při stisknutí tlačítka *Odstranit* (obrázek 4-5) je důležité, aby daný produkt nebyl součástí žádných objednávek nebo dodávek, protože by došlo k významnému

narušení konzistence jednotlivých dat. Proto je tato situace ošetřena a takový produkt není možné odstranit.

The screenshot shows a web form titled 'Produkt'. At the top right is a green button 'Přidat produkt'. Below the title is a dropdown menu 'Vyber produkt:' with 'Aloe' selected. The form contains several input fields: 'ID produktu:' with value '21', 'Název:' with value 'Aloe', 'Doporučená cena:' with value '272,00 Kč', and a large text area for 'Popis:' containing a detailed description of Aloe vera. To the right of these fields are three buttons: 'Upravit' (green), 'Odstranit' (green with a dashed border), and 'Tisknout Produkt' (dark grey). At the bottom right is a green button 'Zavřít'.

Obrázek 4-5 Formulář *frmProdukt*, zdroj: vlastní

Při přidávání nového produktu nebo úpravě stávajícího je uživatel odkázán na nové formuláře, kterými jsou *frmPridatProdukt* a *frmUpravitProdukt* (viz obrázek 4-6). Při přidávání nového produktu má uživatel k dispozici tři textová pole, kde vyplní základní informace. Pokud chce uživatel upravit stávající produkt, je k dispozici velice podobný formulář, který umožňuje upravit název, doporučenou cenu a popis.

The image shows two side-by-side web forms. The left form is titled 'Přidat produkt' and has three input fields: 'Název:', 'Dop. Cena:', and 'Popis:'. The right form is titled 'Upravit produkt' and has the same fields, but the 'ID produktu:' field is pre-filled with '21'. The 'Název:' field contains 'Aloe' and the 'Dop. Cena:' field contains '272'. Both forms have 'Přidat' and 'Zavřít' buttons at the bottom left, and 'Upravit' and 'Zavřít' buttons at the bottom right.

Obrázek 4-6 Formuláře *frmPridatProdukt* a *frmUpravitProdukt*, zdroj: vlastní

4.5.3 Formulář frmDodavatel

K přechodu na formulář frmDodavatel slouží tlačítko *Dodavatelé* na navigačním formuláři *frmUvod*. Tento formulář slouží k zobrazení dat o jednotlivých dodavatelích. V rámci formuláře je opět možné jednotlivé záznamy o dodavatelích editovat, mazat, tisknout nebo vytvářet nové.

Na obrázku 4-7 lze vidět zobrazení tohoto formuláře pro konkrétní dodavatelskou firmu *Polendi*. K přepínání mezi dodavateli slouží opět *pole se seznamem*, které je umístěno v horní části obrazovky. Jelikož na tomto formuláři není povolen přímý přístup k jednotlivým *textovým polím*, jsou k dispozici znovu čtyři tlačítka, a to *Upravit*, *Odstranit*, *Přidat dodavatele* a *Tisknout dodavatele*, které umožňují editaci jednotlivých záznamů. V dolní části formuláře je umístěn *podformulář* s výpisem jednotlivých dodávek od dodavatele. Pro návrat na navigační formulář slouží tlačítko v pravém dolním rohu *Zavřít*.

Při stisknutí tlačítka *Odstranit* (obrázek 4-7), je nutné si znovu dávat pozor na vymazání dodavatelů, s kterými již byly uskutečněny dodávky. Tito dodavatelé odstranění být nemůžou, protože je důležité zachovávat data o jednotlivých dodávkách. Tato možnost odstranění je v aplikaci opět ošetřena a proto není možné takové dodavatele odstranit.

The screenshot displays the 'Dodavatel' (Supplier) form. At the top, there is a header bar with the title 'Dodavatel' and a button 'Přidat dodavatele'. Below the header, a dropdown menu 'Vyber dodavatele' is set to 'Polendi'. The form is divided into several sections: 'Základní údaje' (Basic data) with fields for ID, Name, IČO, and DIČ; 'Adresa' (Address) with fields for Street, City, and Postal Code; 'Kontakt' (Contact) with fields for Phone, Mobile, Email, and Fax. To the right of these fields are buttons for 'Upravit' (Edit), 'Odstranit' (Delete), and 'Tisknout dodavatele' (Print supplier). At the bottom, there is a table titled 'Dodávky' (Deliveries) with columns for 'ID dodávky', 'Datum odeslání', 'Datum vyřízení', and 'Počet položek'. The table contains one row of data. A 'Zavřít' (Close) button is located in the bottom right corner.

ID dodávky	Datum odeslání	Datum vyřízení	Počet položek
2	18.4.2015	20.4.2015	2

Obrázek 4-7 Formulář frmDodavatel, zdroj: vlastní

Pro přidání nových dodavatelů, nebo úpravě již existujících jsou vytvořeny speciální formuláře *frmPridatDodavatel* a *frmUpravitDodavatel* (viz obrázek 4-8). Při vytváření nového dodavatele je k dispozici deset textových polí, ale není nutné vyplnit všechny (viz Tabulka 4-2). Pro editaci již existujících dodavatelů je určen formulář *frmUpravitDodavatel* a je možné upravit všechna pole s výjimkou *ID dodavatel*, které se automaticky vygeneruje samo po stisknutí tlačítka *Přidat* na formuláři *frmPridatDodavatel* a dále s ním již není možné pracovat.

Přidat dodavatele	Upravit dodavatele
Základní údaje	Základní údaje:
Název: <input type="text"/>	ID dodavatele: <input type="text" value="4"/>
IČO: <input type="text"/>	Název: <input type="text" value="Polendi"/>
DIČ: <input type="text"/>	IČO: <input type="text" value="55698942"/>
Adresa:	DIČ: <input type="text" value="PL5569894211"/>
Ulice: <input type="text"/>	Adresa:
Město: <input type="text"/>	Ulice: <input type="text" value="Adamova 777"/>
PSČ: <input type="text"/>	Město: <input type="text" value="Wisla Krakow"/>
Kontakt:	PSČ: <input type="text" value="777 55"/>
Telefon: <input type="text"/>	Kontakt:
Mobil: <input type="text"/>	Telefon: <input type="text" value="558623780"/>
E-mail: <input type="text"/>	Mobil: <input type="text" value="+420 733 555 666"/>
Fax: <input type="text"/>	E-mail: <input type="text" value="polendi@gmail.com"/>
<input type="button" value="Přidat"/> <input type="button" value="Zavřít"/>	Fax: <input type="text" value="558648262"/>
	<input type="button" value="Upravit"/> <input type="button" value="Zavřít"/>

Obrázek 4-8 Formuláře *frmPridatDodavatel* a *frmUpravitDodavatel*, zdroj: vlastní

4.5.4 Formulář *frmOdberatel*

Po stisknutí tlačítka *Odběratelé* na navigačním formuláři *frmUvod* se dostaneme na tento formulář. Slouží k zobrazení informací o jednotlivých odběratelích a prostřednictvím tohoto formuláře lze odběratele upravovat, mazat, vytvářet nové nebo si je jen prohlížet.

Na obrázku 4-9 se nachází formulář *frmOdberatel* aktuálně zobrazený pro odběratele *drMax*. Již na první pohled je patrné, že formulář *frmDodavatel* (obrázek 4-7) a formulář *frmOdberatel* se téměř neliší. Hlavní změnou u tohoto formuláře je umístění jednotlivých objednávek pro konkrétního odběratele prostřednictvím *podformuláře* v dolní části formuláře. Pro každou objednávku je také uvedena cena a tak lze jednoduše filtrovat a zjistit, který odběratel utratí nejvíce.

Pro přístup k jednotlivým údajům jsou určena stejná tlačítka jako u formuláře *frmDodavatel* a u odstranění záznamů platí už několikrát zmíněná pravidla. Pro

přidávání nebo upravování záznamů jsou k dispozici tlačítka *Přidat odběratele* a *Upravit*, které odkazují na nové formuláře *frmPridatOdberatel* a *frmUpravitDodavatel*. Tyto formuláře vypadají téměř totožně s formuláři *frmPridatDodavatel* a *frmUpravitDodavatel* (obrázek 4-8) a proto již zde nejsou uvedeny.

Odběratel: Přidat odběratele

Vyber odběratele: drMax

Základní údaje

ID odběratele: 2 Upravit

Název: drMax

IČO: 54845448

DIČ: bg1111111111 Odstranit

Adresa:

Ulice: Amosova Tisknout odběratele

Město: Praha

PSČ: 45555

Kontakt:

Telefon: 55869898

Mobil: 733666999

E-mail: dr_Max@centrum.cz

Fax: 558999888

Objednávky:

ID objednávky	Datum přijetí	Datum vyřízení	Počet typů	Celková cena (včetně DPH)
1	13.4.2015	16.4.2015	4	3 979,00 Kč
3	15.4.2015	20.4.2015	3	9 717,50 Kč
4	14.4.2015	20.4.2015	2	3 967,50 Kč
8	20.4.2015		2	22 885,00 Kč

Zavřít

Obrázek 4-9 Formulář *frmOdberatel*, zdroj: vlastní

4.5.5 Formulář *frmObjednavka*

Pro přechod na tento formulář slouží tlačítko *Objednávky* na navigačním formuláři *frmUvod*. Prostřednictvím tohoto formuláře uživatel eviduje všechny informace o jednotlivých objednávkách od odběratelů.

Na obrázku 4-10 je zobrazena konkrétní objednávka od odběratele drMax. V horní části formuláře je tlačítko *Přidat objednávku*, které slouží k vytvoření nové objednávky. Formulář je rozdělen do dvou částí, z nichž jedna slouží k manipulaci s objednávkou samotnou a druhá je určena pro jednotlivé položky dané objednávky.

Objednávku můžeme označit jako vyřízenou pomocí tlačítka *Potvrdit objednávku*, což zabezpečí, že daná objednávka už nepůjde žádným způsobem editovat nebo odstranit a poskytne možnost vytisknout dodací list prostřednictvím

tlačítka *Tisknout dodací list*. Objednávku lze jednoduše odstranit, pokud nebyla označena jako vyřízená.

Objednávky Přidat objednávku

ID Objednávky: Potvrdit objednávku

ID Odběratele: Odstranit objednávku

Firma: Upravit objednávku

Datum přijetí: Tisknout dodací list

Datum vyřízení:

◀ ▶

Položky objednávky: Editovat Položky Objednávky

Kód zboží	Název	Počet kusů	Cena za kus	Cena celkem (bez DPH)
1	Goji	100	1,00 Kč	100,00 Kč
4	Artyčok	10	10,00 Kč	100,00 Kč
7	Černá Ředkev	15	100,00 Kč	1 500,00 Kč
10	Hepahit	20	88,00 Kč	1 760,00 Kč

ID objednávky: Cena objednávky (bez dph): 3 460,00 Kč

DPH: 519,00 Kč

Cena objednávky (včetně DPH): 3 979,00 Kč

Zavřít

Obrázek 4-10 Formulář *frmObjednavka*, zdroj: vlastní

K vytvoření nové objednávky nebo úpravě již existující jsou k dispozici tlačítka *Přidat Objednávku* a *Upravit objednávku*, které odkazují na formuláře *frmPridatObjednavku* a *frmUpravitObjednavku* (obrázek 4-11). Při vytváření nové objednávky je k dispozici *pole se seznamem*, pomocí kterého vybíráme konkrétního odběratele. Dále je nutné zadat, kdy jsme objednávku přijali. Jak již bylo zmíněno, formulář *frmUpravitObjednavku* je dostupný jen tehdy, když ještě nebyla objednávka vyřízena. Editace jednotlivých záznamů je pak obdobná jako u vytváření nové objednávky.

Přidat objednávku

ID Odběratele:

Datum přijetí:

Přidat Zavřít

Upravit objednávku

ID Objednávky:

ID Odběratele:

Datum přijetí:

Upravit Zavřít

Obrázek 4-11 Formuláře *frmPridatObjednavku* a *frmUpravitObjednavku*, zdroj: vlastní

Jak již bylo zmíněno, v druhé části formuláře *frmObjednavka* (obrázek 4-10) se nachází *podformulář* s informacemi o jednotlivých položkách objednávky. Je zde detailně popsána každá položka objednávky včetně počtu kusů a jednotlivých cen. Ve spodní části je pak spočítána celková cena objednávky včetně DPH. Pro editaci jednotlivých položek slouží tlačítko *Editovat položky objednávky*, které odkazuje na formulář *frmEditovatPolozkyObjednavky* (obrázek 4-12) pokud již nebyla objednávka vyřízena.

Kód produktu	Název produktu	Počet kusů	Cena za kus (bez dph)	Přidat novou položku
1	Goji	25	156,00 Kč	Upravit
3	Aronie	100	160,00 Kč	Upravit
8	Granátové Jablko	50	195,00 Kč	Upravit
9	Graviola	100	180,00 Kč	Upravit
				Upravit

Obrázek 4-12 Formulář *frmEditovatPolozkyObjednavky*, zdroj: vlastní

Na obrázku 4-12 je zobrazen formulář pro editaci objednávky, konkrétně od odběratele drMax. V horní části formuláře jsou uvedeny informace o konkrétní objednávce, kterou uživatel v danou chvíli upravuje. Ve spodní části jsou umístěny konkrétní položky, které je možno přidávat, odebírat nebo upravovat pomocí tlačítek.

Prostřednictvím tlačítek *Přidat novou položku* a *Upravit*, které je umístěné u každé položky lze konkrétní objednávky editovat. Tlačítko, které vypadá, jako červený křížek už intuitivně napovídá svou funkci odebírání jednotlivých položek. Na obrázku 4-13 jsou uvedeny formuláře *frmPridatPolozkuObjednavky* a *frmUpravitPolozkuObjednavky*, které se otevřou po stisku těchto tlačítek. Při přidávání nové položky je třeba vybrat produkt, počet kusů a cenu za kus. *Pole ID Objednavky*

nám pouze ukazuje, s jakou objednávkou aktuálně pracujeme. Pro úpravu konkrétní položky jsou přístupná pouze pole *Počet kusů* a *Cena za kus*.

Přidat položku:	Upravit položku:
ID Objednávky: 8	ID objednávky: 8
Kód produktu: <input type="text"/>	Kód produktu: 1
Počet kusů: <input type="text"/>	Název: Goji
Cena za kus: <input type="text"/>	Počet kusů: 25
	Cena za kus: 156
<input type="button" value="Přidat"/> <input type="button" value="Zavřít"/>	<input type="button" value="Upravit"/> <input type="button" value="Zavřít"/>

Obrázek 4-13 Formulář *frmPridatPolozkuObjednavky* a *frmUpravitPolozkuObjednavky*, zdroj: vlastní

4.5.6 Formulář *frmDodavky*

Pro přechod na tento formulář slouží tlačítko *Dodávky* na hlavním navigačním formuláři *frmUvod*. Formulář slouží pro veškerou evidenci dodávek a jejich položek od jednotlivých dodavatelů.

Na obrázku 4-14 je zobrazen formulář pro konkrétní dodávku od firmy *Polendí*. Na první pohled je patrné, že formuláře *frmObjednavka* a *frmDodavky* jsou velice podobné. V horní části formuláře, ve které se nachází základní informace o jednotlivých dodávkách, přibyl tlačítko *Status dokončení*, které uživatel zaškrtně, pokud již byla dodávka doručena. Nabízí se možnost tisknout objednávací listy pro jednotlivé dodávky prostřednictvím tlačítka *Tisknout objednávací list*.

Jsou zde k dispozici tlačítka pro vytvoření nebo editaci dodávek, které odkazují na formuláře *frmPridatDodavku* a *frmUpravitDodavku*, které vypadají téměř stejně jako formuláře na obrázku 4-11.

V dolní části formuláře se nachází *podformulář* s informacemi o jednotlivých položkách dodávky, které je možno editovat stisknutím tlačítka *Editovat Položky Dodávky*, pokud již nebyla dodávka označena za dokončenou.

Dodávky

Přidat dodávku

Potvrdit dodávku

Odstranit dodávku

Upravit dodávku

Tisknout objednáací list

ID Dodávky:
ID Dodavatele:
Firma:
Datum odeslání:
Datum přijetí:
Status dokončení: ☒

◀ ▶

Položky dodávky:

Editovat Položky Dodávky

Kód zboží	Název	Počet kusů
5	Brusinka	105
8	Granátové Jablko	100
<input type="text"/>	<input type="text"/>	<input type="text"/>

ID dodávky:

Zavřít

Obrázek 4-14 Formulář frmDodavky, zdroj: vlastní

Prostřednictvím tlačítka pro editaci jednotlivých položek se dostaneme na formulář *frmEditovatPolozkyDodavky*, který je opět velice podobný formuláři pro editaci položek objednávky (obrázek 4-12), pomocí něhož lze přidávat, upravovat a mazat položky.

4.5.7 Sestavy

Pro přístup k jednotlivým sestavám je na navigačním formuláři *frmUvod* k dispozici tlačítko *Tisk*, které nás odkáže na formulář *frmTisk* (Obrázek 4-15).

Na obrázku 4-15 je zobrazen formulář pro tisk jednotlivých sestav. K dispozici jsou tři tlačítka, prostřednictvím kterých lze tisknout *Seznam produktů*, *Seznam dodavatelů* a *Seznam odběratelů*.

Aplikace nabízí možnost vytisknout také dodací a objednávkové listy pro jednotlivé objednávky a dodávky pomocí tlačítek *Tisknout dodací list* (Obrázek 4-10) a *Tisknout objednávkový list* (Obrázek 4-14). Ukázka vzorového dodacího a objednávkového listu se nachází v příloze č. 1 a příloze č. 2.

Obrázek 4-15 Formulář frmTisk, zdroj: vlastní

4.6 Uživatelské prostředí a implementace

Na závěr je databázová aplikace převedena do uživatelského prostředí, což znamená, že daný uživatel nemá přístup k jednotlivým databázovým objektům, a aplikaci ovládá pomocí vytvořených formulářů a sestav a nemůže tak narušit správný a bezpečný chod aplikace. Program Microsoft Access nabízí možnosti základního nastavení pro *aktuální databázi*, kde lze nastavit výchozí formulář (*frmUvod*) prostřednictvím kterého je ovládána celá databázová aplikace. Jednotlivé *místní ovládací nabídky*, *navigační tlačítka* a *speciální klávesy* pro běžnou práci tak lze jednoduše zrušit.

Databázová aplikace bude puštěna do provozu zároveň s odevzdáváním této bakalářské práce a budou do ní nality reálné data. Během prvních měsíců používání budou odstraňovány případné nedostatky, a pokud se v praxi osvědčí, bude implementována pro celou firmu.

5 Závěr

Po několika osobních setkáních se živnostníkem z firmy EkoMedica Czech, s.r.o. byly zjištěny základní nedostatky, které je třeba odstranit. Na základě těchto nedostatků a analýze konkrétních požadavků byla navržena a prakticky vytvořena databázová aplikace, která je zaměřena na evidenci jednotlivých objednávek, dodávek a veškerých informacích o odběratelích, dodavatelích a produktu.

Základem pro tuto práci bylo vytvoření konceptuálního modelu, který popisuje jednotlivé klíčové entity a vztahy, které jsou pro vytvoření aplikace nezbytné. Konceptuální model byl následně převeden do logického, kde jednotlivé entity dostaly podobu tabulek a byly vytvořeny cizí klíče. Z logického modelu pak vychází cílová implementace do modelu fyzického v rámci prostředí Microsoft Access.

V prostředí programu Microsoft Access byl vytvořen podrobný návrh jednotlivých tabulek, do kterých se budou ukládat všechna data. Pro vytvoření uživatelského rozhraní byly použity formuláře, kde správnou funkci a jejich vzájemnou integraci zajišťuje programovací jazyk VBA. Posledním krokem bylo vytvořit konkrétní sestavy pro tisk určených dokumentů, jakou jsou dodací listy, objednávkové listy, a různé seznamy. Aplikace byla uložena na CD.

Cílem této bakalářské práce bylo vytvořit databázovou aplikaci v programu Microsoft Access, která by měla plnit následující funkce:

- zaznamenávání veškerých údajů o zboží,
- správa dodacího listu a objednávkového formuláře,
- řízení, správa a evidence jednotlivých objednávek a dodávek,
- zaznamenání veškerých údajů o odběratelích a dodavatelích,
- jednoduchý grafický design s intuitivním ovládáním.

Všechny výše uvedené funkce databázová aplikace splňuje a bude nasazena do provozu zároveň s odevzdáváním této bakalářské práce. Pokud se v praxi osvědčí, bude následně implementována pro celou firmu. V budoucnu by mohla být rozšířena o skladovou evidenci a fakturační systém.

Seznam použité literatury

Tištěné zdroje:

OPPEL, Andrew J. *Databáze bez předchozích znalostí*. Brno: Computer Press, 2006. ISBN 80-251-1199-7.

OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.

CONNOLLY, T., C. BEGG a R HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.

KRUCZEK, Aleš. *Microsoft Access 2010: podrobná uživatelská příručka*. Brno: Computer Press, 2010. ISBN 978-80-251-3289-0.

SHEPHER, Richard. *Access VBA: výukový průvodce*. Brno: Computer Press, 2012. ISBN 978-80-251-3686-7.

RIORDAN, Rebecca. *Vytváříme relační databázové aplikace*. Praha: Computer Press, 2000. ISBN 80-7226-360-9.

Elektronické zdroje:

NOVÁK, Vítězslav. Přednášky předmětu: Základy databází [online]. 2014. [cit. 2015–04–05]. Dostupné z: <http://lms.vsb.cz/course/view.php?id=8933>

Seznam zkratek

1NF – První normální forma

2NF – Druhá normální forma

3NF – Třetí normální forma

4NF – Čtvrtá normální forma

5NF – Pátá normální forma

BCNF – Boyce Coddova normální forma

DBMS – Database Management Systém

DCL – Data Control Language

DDL – Data Definition Language

DIČ – Daňové identifikační číslo

DML – Data Manipulation Language

DPH – Daň z přidané hodnoty

ER – Entitně relační

FK – Cizí klíč / Foreign Key

IČO – Identifikační číslo osoby

ODBC – Open Database Connectivity

PK – Primární klíč / Primary key

PSC – Poštovní směrovací číslo

SQL – Structured Query Language

SŘBD – Systém řízení báze dat

SW – Software

TCC – Transaction Control Commands

VBA – Visual Basic for Application

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, bakalářskou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

Ve Frýdku - Místku dne 7. 5. 2015



Aleš Miňo

Seznam příloh

- Příloha č. 1 – Ukázka konkrétního dodacího listu
- Příloha č. 2 – Ukázka konkrétního objednávkového listu
- Příloha č. 3 – CD s databázovou aplikací

Přílohy

Příloha č. 1

Dodavatel:

EkoMedica Czech, s.r.o.

Sokolská třída 1615/50

702 00 Ostrava

29292594

CZ29292594

**Odběratel:**

drMax

Amosova

45555 Praha

IČO: 54845448

DIČ: bg1111111111

Objednávka - Dodací list - Číslo: 1 /2015

Název	Počet kusů	Cena za kus (Bez DPH)	Cena za kus (S DPH)	Doporučená prodejní cena
Goji	100	1,00 Kč	1,15 Kč	299,00 Kč
Artyčok	10	10,00 Kč	11,50 Kč	338,00 Kč
Černá Ředkev	15	100,00 Kč	115,00 Kč	338,00 Kč
Hepahit	20	88,00 Kč	101,20 Kč	199,00 Kč

Datum: 16.4.2015

Příloha č. 2

Objednávka číslo: 2

ze dne: 18.4.2015

Termín dodání: 20.4.2015

Plátce DPH: ANO - NE

Přeprava:

Odběratel:

EkoMedica Czech,
Sokolská třída 161
702 00 Ostrava

IČO: 29292594
DIČ: CZ29292594

Dodavatel:

Polendi
Adamova 777
77755 Wisla Krakow

IČO: 55698942
DIČ: PL5569894211

ID Položky	Počet kusů	Název zboží
1	25	Goji
3	70	Aronie
5	105	Brusinka
8	100	Granátové Jablko

	Razítko a podpis:
--	-------------------